



Diogo Alexandre de Jesus Ferreira

Licenciado em Ciências de Engenharia Eletrotécnica e de Computadores

Plataforma IoT Multi-sensor para Monitorização Integrada de Materiais PCM

Dissertação para obtenção do Grau de Mestre em
Engenharia Eletrotécnica e de Computadores

Orientador: João Pedro Abreu de Oliveira, Professor Doutor,
Universidade Nova de Lisboa

Co-orientador: João Miguel Murta Pina, Professor Doutor,
Universidade Nova de Lisboa

Júri

Presidente: Prof. Doutor Paulo Miguel de Araújo Borges Montezuma de Carvalho
Arguente: Prof. Doutor Rui Miguel Amaral Lopes
Vogal: Prof. Doutor João Pedro Abreu de Oliveira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Fevereiro, 2020

Plataforma IoT Multi-sensor para Monitorização Integrada de Materiais PCM

Copyright © Diogo Alexandre de Jesus Ferreira, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Para Orlando Ferreira

AGRADECIMENTOS

Primeiramente, quero agradecer ao meu orientador Professor João Pedro Oliveira e ao meu coorientador João Murta Pina pela disponibilidade demonstrada e auxílio oferecido durante o decorrer do projecto da dissertação.

Não posso deixar de agradecer à Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia e a todo o corpo docente que me acompanhou no meu percurso académico, proporcionando um grande crescimento tanto a nível académico como a nível pessoal.

Gostaria de agradecer aos meus amigos Pedro Boto e Rui Martins que me acompanharam neste percurso e que o tornaram bastante divertido e sempre cheio de momentos que irei recordar. Entre estes momentos, gostaria de mencionar os almoços no "forum" ao qual, ainda havemos de ir lá outra vez.

Um muito obrigado também, a todos os que integram o meu grupo de amigos e que de alguma forma contribuíram tanto para a realização do meu percurso académico como para a realização da minha dissertação, salientando um obrigado ao Gonçalo Biscaia e ao André Antunes.

Por ultimo, um agradecimento muito especial à minha família, especialmente aos meus pais e irmã, por todo o apoio, carinho, dedicação e paciência que têm demonstrado na realização deste percurso académico.

A todos,
Muito Obrigado!

*"The most important thing is to keep the most important thing
the most important thing."
- Donald P. Coduto*

RESUMO

A eficiência energética é um conceito de grande relevância nos dias de hoje devido à presente crise económica, energética e ambiental. O que era outrora um bom fator de eficiência poderá já não o ser, por isso é necessário pesquisar e implementar novos métodos que permitam melhorar a eficiência energética dos equipamentos. Posto isto, pretende-se com a implementação de *phase change materials* efetuar esse melhoramento, ao armazenar o calor gerado de uma maneira mais pratica e eficiente utilizando propriedades de *thermal energy storage*, no entanto para determinar se a implementação destes materiais é efetivamente benéfica, é imprescindível a projeção de um sistema de monitorização capaz de fornecer as métricas necessárias para o calculo da performance energética de equipamentos antes e depois da introdução de PCM.

Neste sentido, pretende-se com esta dissertação efetuar a projeção e implementação inicial de uma plataforma *internet of things* multi-sensor para a monitorização integrada de PCM. Pretende-se que esta plataforma seja: versátil e modular, de modo a acomodar o maior numero de sistemas que possam ser equipados com PCM; reconfigurável, para que seja possível estabelecer comunicação à plataforma IoT em vários ambientes e que as métricas obtidas se adequem ao sistema monitorizado; por fim, pretende-se que a plataforma possua a capacidade de armazenar e exibir os dados de uma maneira simples para facilitar análises posteriores.

Os resultados obtidos demonstram que a plataforma implementada possui uma performance notável, possuindo a estabilidade necessária para estar longos períodos de tempo ativa e uma precisão semelhante relativamente à verificada em equipamentos comumente utilizados na industria. Porém, verificou-se que, mediante a alteração das condições físicas de funcionamento, o sistema poderia ter comportamentos irregulares, no entanto, é possível melhorar bastante a performance do sistema numa segunda iteração, permitindo assim a monitorização de virtualmente qualquer equipamento.

Palavras-chave: Eficiência Energética, *Phase Change Materials*, *Thermal Energy Storage*, *Internet of Things*, Monitorização, Multi-Sensor.

ABSTRACT

Energy efficiency is a concept of great relevance today due to the current economic, energy and environmental crisis. What was once a good efficiency factor may no longer be, so it is necessary to research and implement new methods to improve the energy efficiency of equipments. That said, it is intended with the implementation of phase change materials to make this improvements, by storing the heat generated in a more practical and efficient way using properties of thermal energy storage, however to determine if the implementation of these materials is indeed beneficial, it is essential to design a monitoring system capable of providing the necessary metrics for calculating the energy performance of the equipment before and after the introduction of PCM.

In this sense, this dissertation intends to carry out the projection and initial implementation of a multi-sensor internet of things platform for the integrated monitoring of PCM. This platform is intended to be: versatile and modular, in order to accommodate the largest number of systems that can be equipped with PCM; reconfigurable, so that it is possible to establish communication to the IoT platform in various environments and that the obtained metrics are adapted to the monitored system; finally, it is intended that the platform has the ability to store and display data in a simple way to facilitate further analysis.

The results obtained demonstrate that the implemented platform has a remarkable performance, having the necessary stability to be active for long periods of time and an accuracy similar to that verified in equipments commonly used in the industry. However, it was found that, by changing the physical operating conditions, the system could behave irregularly, however, it is possible to greatly improve the performance of the system in a second iteration, thus allowing the monitoring of virtually any equipment.

Keywords: Energy efficiency, Phase Change Materials, Thermal Energy Storage, Internet of Things, Monitoring, Multi-Sensor.

ÍNDICE

Lista de Figuras	xvii
Lista de Tabelas	xxi
Listagens	xxiii
Siglas	xxv
1 Introdução	1
1.1 Motivação	1
1.2 Objectivos	1
1.3 Estrutura do Documento	2
2 Revisão da Literatura e Fundamentos Tecnológicos	5
2.1 Conceito de Calor	5
2.1.1 Calor Latente e sensível	6
2.2 Armazenamento de Energia Térmica	7
2.3 Materiais de Mudança de Fase (PCM)	8
2.3.1 Materiais orgânicos	9
2.3.2 Materiais inorgânicos	11
2.3.3 Eutéticos	13
2.4 Métodos de incorporação de PCMs	13
2.4.1 Aplicações	15
2.5 <i>Internet of Things</i> para a Monitorização Integrada de Sistemas	16
2.5.1 Micro-Controladores	17
2.5.2 <i>Analog to Digital Converters</i>	23
2.5.3 Módulos de sensores	25
3 Projeto da Arquitetura e Implementação do Sistema de Monitorização para PCM	29
3.1 Equipamento a utilizar	29
3.2 Arquitetura do Sistema	31
3.3 Módulo de sensores	33
3.3.1 Medição de temperatura	33

3.3.2	Módulo de monitorização de potência	35
3.4	Módulo de aquisição de dados	40
3.4.1	Escolha do MCU	40
3.4.2	Implementação	41
3.5	Módulo de comunicação	43
3.5.1	Implementação	43
3.6	Plataforma IoT	45
3.6.1	Implementação	46
4	Resultados Experimentais	51
4.1	Validação da plataforma	51
4.1.1	Cenário 1 - Teste controlo utilizando uma carga conhecida.	51
4.1.2	Cenário 2 - Monitorização do funcionamento de um frigorífico com- binado	56
4.1.3	Cenário 3 - Monitorização do funcionamento de um jarro elétrico	64
4.1.4	Cenário 4 - Monitorização de uma chocolateira industrial	66
4.2	Ensaio em equipamento com materiais PCM	72
4.2.1	Casos de estudo e montagem do sistema para o ensaio	72
4.2.2	Análise dos dados obtidos	73
5	Conclusão e Trabalhos Futuros	77
5.1	Trabalhos futuros	78
	Bibliografia	81
I	Código utilizado no MCU PSoC 5Lp	85
II	Código utilizado no MCU ESP32	97
III	Query utilizada no serviço <i>Stream Analytics</i>	105

LISTA DE FIGURAS

2.1	Diagrama teórico de mudança de fase Sólido-Líquido.	7
2.2	Diagrama de Armazenamento de Energia Térmica. (Adaptado de ([9])). . . .	8
2.3	Diagrama materiais PCM.	8
2.4	Diagrama representativo da mudança de fase. (Adaptado de [12])).	10
2.5	Painel flexível contendo <i>Phase Change Materials</i> (PCM).	14
2.6	Micro cápsula PCM. (Adaptado de [13])).	15
2.7	Conceitos essenciais para <i>Internet of Things</i> (IoT).	17
2.8	Produtos <i>Arduino</i>	18
2.9	ESP32.	19
2.10	PSoC 5Lp.	20
2.11	Diagrama de blocos PSoC 5Lp, família CY8C58LP. (Adaptado de [25])). . . .	21
2.12	Raspberry Pi modelo 4 B.	22
2.13	Funcionamento SAR ADC.	23
2.14	Fluxograma SAR <i>Logic</i>	24
2.15	Diagrama simplificado ADC Sigma Delta e <i>noise shaping</i>	24
2.16	Análise espectral de ADC Sigma-Delta. (Adaptado de [30])).	25
3.1	Equipamento a utilizar - Chocolateira industrial.	30
3.2	Diagrama de blocos do equipamento	30
3.3	Arquitetura do sistema.	31
3.4	Arquitetura do sistema, agrupamento de módulos.	32
3.5	Arquitetura do sistema diferentes comunicações.	32
3.6	Ligação entre o sensor e o Microcontrolador em modo <i>Single-drop</i>	34
3.7	Implementação de sensores DS18B20 com recurso a um bloco de interface pré-programado.	34
3.8	Fluxograma referente ao funcionamento do bloco utilizado para fazer a interface com os sensores DS18B20.	35
3.9	Diagrama de blocos funcional referente ao ic ACS71020.	36
3.10	Divisor de tensão para leitura de tensão utilizando ACS71020.	37
3.11	Bloco de comunicação SPI com tensão de referência de 3,3V gerada por DAC de tensão.	39
3.12	Bloco I2C slave presente no PSoC 5Lp.	41

3.13 Fluxograma referente ao comportamento do módulo de aquisição de dados. .	42
3.14 Fluxograma do funcionamento do módulo de comunicação.	44
3.15 Serviços utilizados na plataforma IoT.	46
3.16 Criação do serviço IoT Hub.	47
3.17 Chave de acesso primária.	47
3.18 Criação de uma conta de armazenamento.	48
3.19 Definições detalhadas sobre a adição dos serviços <i>IoT Hub</i> e <i>Blob Storage</i> ao <i>Stream Analytics</i>	49
4.1 Módulo de monitorização de potência. (A) - Alimentação 3,3v, (B) - Comunicação e alimentação, (C) - Divisor resistivo, (D) - Bloco de terminais para Fase e Neutro.	52
4.2 Montagem completa. Alimentação a partir de um <i>Power Bank</i>	52
4.3 Medidor de potência PM9C da <i>Schneider Electric</i>	53
4.4 Lâmpada de incandescência de 70W com sensor de temperatura.	53
4.5 Consumo energético e temperatura de uma lâmpada de incandescência de 70W em função do tempo.	54
4.6 Esquema termodinâmico do frigorífico combinado. (1)(2) - Compressores, (3) - Unidade condensadora, (4)(5) - Unidades evaporadoras.	57
4.7 Perfil de carga de um frigorífico, congelador e combinados.	58
4.8 Consumo energético e temperatura de um frigorífico combinado em função do tempo. Consumo estável, seguido de um período sem consumo e um arranque do ciclo de refrigeração do congelador.	59
4.9 Consumo energético e temperatura de um frigorífico combinado em função do tempo. Arranque de ambos os ciclos de refrigeração.	60
4.10 Consumo energético e temperatura de um frigorífico combinado em função do tempo. Arranque de ambos os ciclos de refrigeração seguida de uma única abertura da porta do frigorífico.	62
4.11 Consumo energético e temperatura de um frigorífico combinado em função do tempo. Uso quotidiano.	63
4.12 Teste de monitorização de um jarro elétrico.	65
4.13 Chocolateira industrial equipada com sensores de temperatura e monitorizador de potência consumida.	66
4.14 Teste de monitorização de uma chocolateira industrial de 3000W.	67
4.15 Esquema representando o equipamento, sistema de monitorização e problema devido ao isolamento do medidor de potência.	69
4.16 Simulação problema de isolamento, queda de tensão eficaz em função da resistência à massa do sensor de temperatura.	70
4.17 Teste de monitorização de uma chocolateira industrial de 3000W com valores de tensão e fator de potência corrigidos.	71

4.18 Esquerda para a direita: mistura de chocolate e leite, PCM (S34) encapsulado e PCM (S34) aplicado na chocolateira.	73
4.19 Ensaio de controlo utilizando água e uma mistura de chocolate e leite. Carga e descarga com armazenamento térmico sensível.	74
4.20 Ensaio utilizando PCM e uma mistura de chocolate e leite. Carga e descarga com armazenamento térmico latente.	75

LISTA DE TABELAS

2.1	Comparação entre tipos de PCMs. (Adaptado de [10]).	9
2.2	Características de Ceras de Parafina Puras. (Adaptado de [13]).	10
2.3	Características de Ácidos Gordos. (Adaptado de [7].)	11
2.4	Características de Glicóis. (Adaptado de [15]).	11
2.5	Características de metais. (Adaptado de [14]).	12
2.6	Características Hidratos de Sal. (Adaptado de [14]).	13
2.7	Características Eutéticos. (Adaptado de [14]).	13
2.8	Comparação dos Micro-controladores.	22
2.9	Sensores de temperatura.	26
2.10	Módulos medição de energia eléctrica.	27
3.1	Registos do ic ACS71020, grandezas correspondentes e fator de escala	39

LISTAGENS

3.1	Exemplo de medição de temperaturas.	35
3.2	Rotinas de escrita e leitura I2c.	43
I.1	Código PSoC 5Lp.	85
II.1	Código ESP32.	97
III.1	<i>Query</i> utilizada no serviço <i>Stream Analytics</i>	105

SIGLAS

ADC	<i>Analog to Digital Converter.</i>
AVAC	<i>Aquecimento Ventilação e Ar Condicionado.</i>
BLE	<i>Bluetooth Low Energy.</i>
CEE	<i>Conservação Energética e Eficiência.</i>
CLK	<i>Clock.</i>
CS	<i>Chip-Select.</i>
CSV	<i>Comma-separated values.</i>
DAC	<i>Digital to Analog Converter.</i>
DNL	<i>Differential Nonlinearity.</i>
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory.</i>
I2C	<i>Inter-Integrated Circuit.</i>
IC	<i>Integrated Circuit.</i>
IDE	<i>Integrated Development Environment.</i>
INL	<i>Integral Nonlinearity.</i>
IoT	<i>Internet of Things.</i>
JSON	<i>JavaScript Object Notation.</i>
LSB	<i>Least Significant Bit.</i>
MCU	<i>Microcontroller Unit.</i>
MQTT	<i>Message Queuing Telemetry Transport.</i>
MQTT	<i>MQ Telemetry Transport.</i>

NFC	<i>Near Field Communication.</i>
OpAmp	<i>Operational Amplifier.</i>
PCM	<i>Phase Change Materials.</i>
PSoC	<i>Programmable System-on-Chip.</i>
PWM	<i>Pulse Width Modulation.</i>
RAM	<i>Random Access Memory.</i>
RFID	<i>Radio-Frequency Identification.</i>
RTC	<i>Real Time Clock.</i>
RTD	<i>Resistance Temperature Detector.</i>
S&H	<i>Sample and Hold.</i>
SAR	<i>Successive Approximation Register.</i>
SCL	<i>Serial Clock.</i>
SDA	<i>Serial Data.</i>
SPI	<i>Serial Peripheral Interface.</i>
SQL	<i>Structured Query Language.</i>
TES	<i>Thermal Energy Storage.</i>
UART	<i>Universal Asynchronous Receiver-Transmitter.</i>
UDB	<i>Universal Digital Blocks.</i>
UL	<i>Underwriters Laboratories.</i>
USB	<i>Universal Serial Bus.</i>
WISP	<i>Wireless Identification and Sensing Platforms.</i>
WSAN	<i>Wireless Sensor and Actuator Networks.</i>

INTRODUÇÃO

1.1 Motivação

Nos tempos de crise económica, energética e ambiental a Conservação Energética e Eficiência (CEE) são conceitos chave para o sector energético [1]. Os combustíveis fósseis têm vindo a servir e cumprir todas as necessidades humanas ao longo desta era, no entanto, ao mesmo tempo causaram os danos ambientais que vieram a levar aos mais emergentes problemas ambientais com os quais lidamos. Sendo assim, é necessário desenvolver métodos que visam o melhoramento da eficiência energética, de maneira a que a energia não seja consumida de uma maneira irresponsável e com desperdício.

Com o intuito de dar mais um passo em direção à conservação e eficiência energética, esta dissertação focar-se-á na criação de uma plataforma *Internet of Things* (IoT) multi-sensor para monitorização de sistemas que façam uso de *Phase Change Materials* (PCM), sendo que a utilização desses mesmos materiais em sistemas térmicos é uma das soluções com maior potencial na conservação e utilização eficiente de energia térmica, nomeadamente o calor latente.

PCMs são substâncias que libertam ou absorvem energia (calor latente e sensível) no processo de mudança de fase, de modo a fornecer um aquecimento ou arrefecimento de uma maneira mais eficiente. Calor pode ser armazenado na forma de calor sensível e ou latente. O armazenamento de calor latente é normalmente usado com transições de fase sólido-líquido devido a limitações volumétricas das instalações [2].

1.2 Objectivos

Esta dissertação terá como objetivos principais os seguintes pontos:

- Apresentar o tema da necessidade de conservação e armazenamento de energia;

- Analisar os diferentes tipos de armazenamento de energia térmica e os tipos de PCM existentes, orgânicos, inorgânicos e eutéticos;
- Referir os possíveis tipos de incorporação¹ de PCMs;
- Apresentar o conceito de IoT;
- Estudar e comparar os micro-controladores, sensores e plataformas IoT existentes no mercado;
- Analisar as diferentes características, vantagens e desvantagens dos diferentes tipos de sensores utilizados;
- Descrever todo o trabalho realizado para a criação e implementação de uma Plataforma IoT Multi-Sensor para Monitorização Integrada de PCM.

Gostaria também de se mencionar que este trabalho foi possível, em parte, pela colaboração da empresa METALOGONDE Indústria Metalomecânica Lda. a quem se agradece a contribuição para o protótipo experimental.

1.3 Estrutura do Documento

A presente dissertação possui diferentes capítulos com o intuito de levar o leitor a uma explicação do trabalho realizado encontrando-se estruturado em cinco capítulos, sendo estes:

- Introdução;
- Revisão da Literatura e Fundamentos Tecnológicos;
- Projeto da Arquitetura e do Sistema de Monitorização para PCM;
- Resultados Experimentais;
- Conclusão e Trabalhos Futuros.

No capítulo **Revisão da Literatura e Fundamentos Tecnológicos**, é realizada uma leitura aprofundada acerca dos temas chave relacionados com esta dissertação. Primeiramente serão apresentados os diferentes conceitos de energia térmica necessários para a compreensão do funcionamento e das características dos PCM, após realizado um estudo sobre os materiais serão abordados os métodos atuais de incorporação dos mesmos. De seguida, será abordado o tema referente ao IoT onde será feita uma pequena análise do mesmo seguida de alguns exemplos de práticos. Posteriormente, será feita uma análise de mercado onde se pretende determinar quais serão os *Microcontroller Unit* (MCU) e

¹Técnicas utilizadas para inserir o PCM em outros materiais ou meios.

sensores que se destacam no âmbito da monitorização, para que possam ser utilizados no sistema a implementar.

O capítulo **Projeto da Arquitetura e do Sistema de Monitorização para PCM**, consiste na explicação das decisões tomadas para a criação da arquitetura e para a escolha, dimensionamento e implementação dos diferentes sensores, MCU e plataformas IoT disponíveis.

De seguida, no capítulo **Resultados Experimentais**, são introduzidos os diversos cenários utilizados para o teste de características específicas do sistema de monitorização implementado. Estes cenários focam-se primeiramente no teste do sistema de monitorização em si, avaliando a precisão dos dados obtidos e a estabilidade de funcionamento em longos períodos de monitorização. Por fim, serão efetuados ensaios experimentais onde se pretende testar o funcionamento de um equipamento que utilize armazenamento térmico latente. Após determinado o seu perfil de funcionamento, esse mesmo equipamento foi equipado com PCM, passando efetivamente para um armazenamento térmico sensível. Tendo um ponto de comparação os perfis de funcionamento são comparados, de modo a determinar os benefícios e desvantagens da utilização de materiais PCM para o armazenamento térmico.

Por fim, no capítulo **Conclusões e Trabalhos Futuros**, é averiguado se foi possível corresponder às características de implementação desejadas e é formulada uma conclusão à cerca de todo o trabalho desenvolvido, apresentando no final alternativas, desde otimizações ao sistema desenvolvido, até novas implementações de serviços e funções que se consideram úteis para o desenvolvimento de uma ferramenta capaz de ser utilizada em todo o tipo de equipamentos, condições e funções.

REVISÃO DA LITERATURA E FUNDAMENTOS TECNOLÓGICOS

Pretende-se neste capítulo efectuar um estudo sobre os (PCM) e a sua monitorização via (IoT). Primeiramente, serão abordados os conceitos fundamentais de armazenamento de energia térmica transitando para o estudo das características e funcionamento dos PCM. Seguidamente, serão abordados os temas referentes ao IoT, analisando também os equipamentos que poderão ser utilizados para esta aplicação.

2.1 Conceito de Calor

Para que se entenda realmente o que é calor latente e sensível é necessário primeiro compreender o conceito de calor, mais propriamente, energia térmica. Para tal, é habitual fazer uma pequena analogia entre o conceito de trabalho e energia cinética.

O conceito de trabalho está normalmente associado ao deslocamento de um sistema na presença de uma força, que atua sobre o mesmo [3]. Sendo assim, trabalho é definido pelo produto entre a força e o deslocamento. O sistema desloca-se a uma determinada velocidade, fazendo assim com que cada vez que é aplicado trabalho ao sistema este tem a sua energia cinética alterada [4]. Analogamente, é possível associar energia térmica ao trabalho gerado pelos átomos pertencentes a um sistema, visto que estes possuem energia cinética. É de notar que o conceito de temperatura e de calor são distintos. O calor corresponde à energia transferida devido a uma diferença de temperatura [3]. A temperatura é proporcional à energia cinética média dos átomos e moléculas presentes no sistema(entropia) [4].

2.1.1 Calor Latente e sensível

O foco principal do presente estudo é o armazenamento de calor latente e sensível, sendo que ambos são componentes da energia interna de um sistema ($U_{latente}$) e ($U_{sensível}$) respectivamente, e que em conjunto são referidos como energia térmica (U_t) [5]. Esta energia térmica pode ser armazenada usando processos, químicos ou físicos, sendo que cada um possui características distintas. Relativamente aos tipos de calor, calor sensível é a porção de energia térmica utilizada para que um sistema mude de temperatura. Adicionalmente, calor latente é a porção de energia térmica, absorvida ou libertada, durante uma mudança de fase de um material. Na Figura 2.1 é possível observar a diferença entre calor sensível e latente presente numa mudança de fase. Existem três mudanças de fase possíveis:

- **Líquido-gás:** Esta mudança de fase está normalmente associada a quantidades de energia superiores relativamente às outras mudanças de fase descritas; Este processo é fortemente dependente das condições do meio, nomeadamente, volume constante, pressão constante em sistemas fechados e, por último, pressão constante em sistemas abertos [6];
- **Sólido-Líquido:** Neste processo de mudança de fase ocorre normalmente uma pequena mudança de volume, cerca de 10%. Se o recipiente estiver adequadamente dimensionado à fase com maior volume, normalmente líquida, não existirá uma grande mudança de pressão [6]. Sendo este um processo isotérmico¹ numa pressão praticamente constante (onde se encontram os PCMs mais comuns) esta mudança de fase é uma das mais adequadas para aplicações técnicas. Fazendo também com que seja possível selecionar um PCM, com base na temperatura de fusão do mesmo [7];
- **Sólido-Sólido:** Este tipo de mudança de fase apresenta características semelhantes à mudança de fase Sólido-Líquido, no entanto não possui um balanço energético tão expressivo [6].

¹ Transformação termodinâmica que ocorre a temperatura constante num sistema fechado, sistema este que permite trocas de energia com o exterior, mas não de matéria.

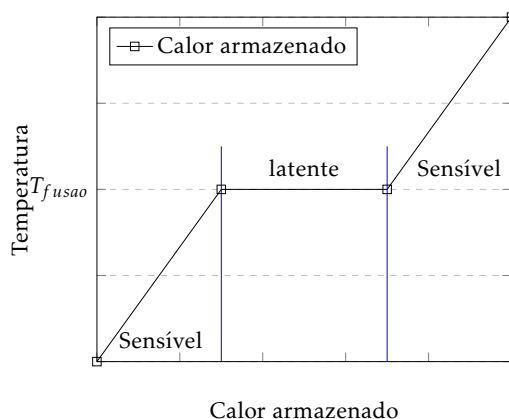


Figura 2.1: Diagrama teórico de mudança de fase Sólido-Líquido.

2.2 Armazenamento de Energia Térmica

Com a utilização de sistemas de *Thermal Energy Storage* (TES), é possível efectuar o armazenamento de energia que outrora seria desperdiçada. É também possível evitar o consumo excessivo de energia, devido ao facto de existir à partida energia armazenada e pronta a ser utilizada, fazendo com que a necessidade do consumo de energia proveniente de outras fontes seja reduzida. O armazenamento de energia térmica é um processo de conservação de energia temporário, onde se armazena calor para uma utilização futura [8]. Alguns exemplos de TES são:

- Acumulação de energia solar para utilização à noite;
- Armazenamento de calor nas horas de baixa tarifa eléctrica, para uso posterior em horas de tarifa alta;
- Utilização de baterias térmicas para alimentação de sistemas;

Na Figura 2.2 é possível observar um diagrama que ilustra os diferentes tipos de armazenamento térmico.

No processo de armazenamento de energia térmica existem duas opções seleccionáveis. A primeira consiste no aquecimento ou arrefecimento de uma substância de modo a inserir ou retirar **calor sensível** do sistema em causa. A segunda opção consiste na alteração de fase de uma substância, nomeadamente PCM, onde o material poderá sofrer solidificação, fusão ou vaporização sendo estes os processos onde se armazenará **calor latente** ou se libertará o mesmo caso ocorram os processos inversos.

Os sistemas de armazenamento de calor sensível não são ser estudados neste trabalho. Porém, os sistemas de armazenamento de calor latente e os seus materiais, PCM, são estudados com mais detalhe nas secções seguintes.

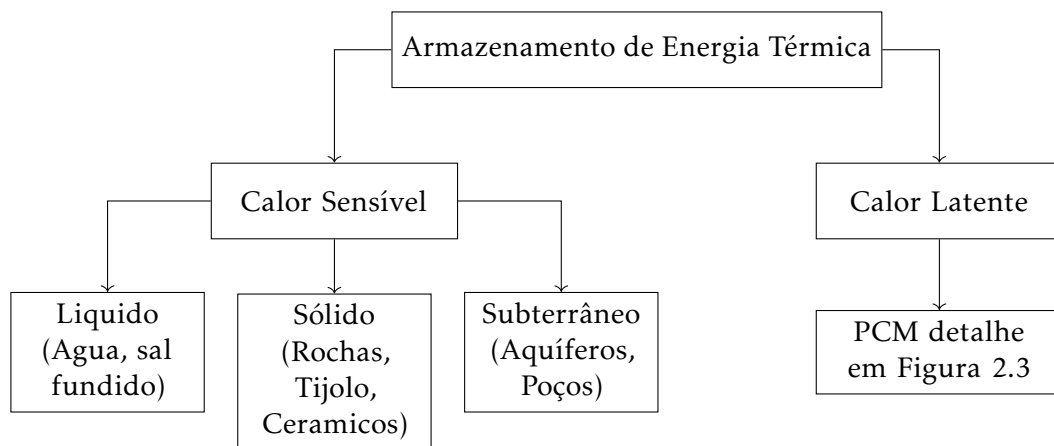


Figura 2.2: Diagrama de Armazenamento de Energia Térmica. (Adaptado de ([9])).

2.3 Materiais de Mudança de Fase (PCM)

Com base nas propriedades de mudança de fase dos materiais, a tipificação é composta por três categorias distintas: PCMs sólido-sólido, PCMs sólido-líquido e PCMs líquido-gás. Dentro da categoria sólido-líquido encontram-se os PCMs mais adequados para TES. Nesta categoria, sólido-líquido, encontram-se três grupos distintos de PCMs os orgânicos, inorgânicos e eutéticos [10]. Na figura 2.3 é possível observar um diagrama que organiza os diferentes tipos de PCMs bem como o material constituinte.

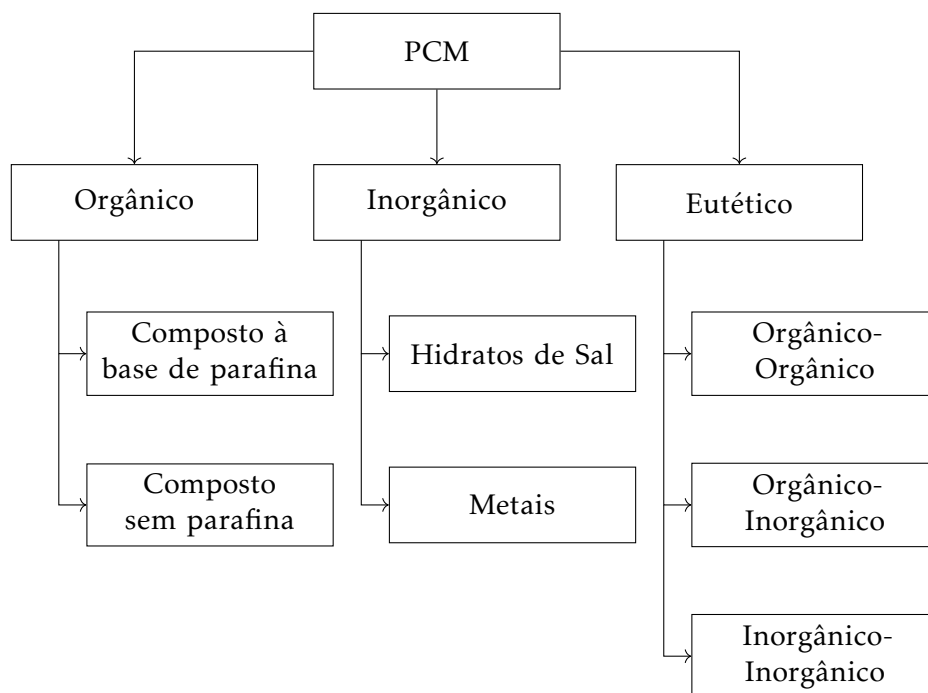


Figura 2.3: Diagrama materiais PCM.

De modo a compreender o funcionamento destes materiais, é necessário ter em conta algumas propriedades que caracterizam os PCMs e que ajudarão, mais à frente, a fazer uma comparação entre PCMs diferentes. Estas propriedades são o calor latente de fusão (ΔH_{fus}), temperatura de fusão (T_{fus}) e condutividade térmica (K) [7].

A tabela 2.1 resume as vantagens e desvantagens de cada tipo de PCM.

Tabela 2.1: Comparação entre tipos de PCMs. (Adaptado de [10]).

Tipo	Vantagens	Desvantagens
Orgânicos	1. Disponibilidade em várias temperaturas de fusão. 2. Alta Temperatura de fusão. 3. Ausência de super-arrefecimento. 4. Quimicamente estável e reciclável 5. Boa compatibilidade com outros materiais.	1. Baixa condutividade térmica. 2. Grande alteração de volume relativo. 3. Inflamável.
Inorgânicos	1. Alta Temperatura de fusão. 2. Alta condutividade térmica. 3. Pequena alteração de volume relativo. 4. Disponível a baixo custo.	1. Super-arrefecimento. 2. Corrosão.
Eutéticos	1. Temperatura de fusão altamente precisa. 2. Alta densidade volumétrica de armazenamento térmico.	1. Falta de dados sobre propriedades termo-físicas.

De acordo com [11], é possível observar três comportamentos distintos aquando da fusão:

- **Fusão Congruente:** Dada uma temperatura específica (T_{fus}) é possível que estes materiais fundam e solidifiquem sem que exista uma segregação de fase ou degradação do seu ΔH_{fus} ;
- **Fusão Incongruente:** Ocorre quando o sal não é inteiramente solúvel na água. Existe elevada segregação de fase;
- **Fusão Semi-Congruente:** Apresenta um comportamento intermédio entre a fusão congruente e incongruente. Existe segregação de fase durante o processo de fusão;

Na figura 2.4 é possível verificar que o PCM ao receber calor inicia o processo de fusão e ao emitir calor inicia o seu processo de solidificação.

2.3.1 Materiais orgânicos

Os materiais orgânicos podem ser divididos em duas subcategorias: compostos à **base de parafina** e compostos **sem parafina**, figura 2.3. Estes materiais que possuem propriedades de fusão congruente, têm grande estabilidade térmica, estão disponíveis em várias

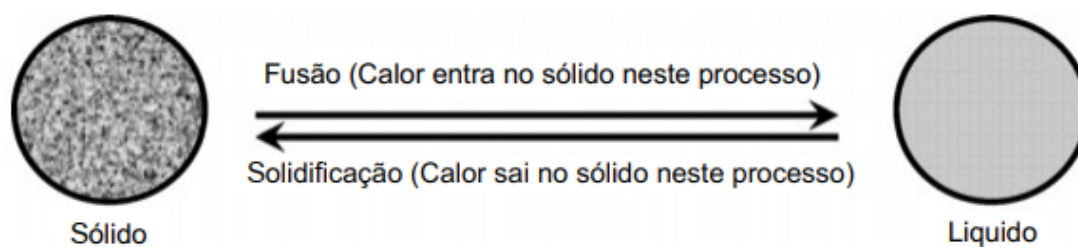


Figura 2.4: Diagrama representativo da mudança de fase. (Adaptado de [12]).

temperaturas de fusão e estão ausentes de super-arrefecimento². Ver tabela 2.1 para mais detalhes.

2.3.1.1 Compostos à Base de Parafina

As parafinas utilizadas como PCM para TES são normalmente mencionadas como ceras de parafina, compostas maioritariamente por uma cadeia de n-alcanos em linha. É necessário salientar que ceras de parafina puras são bastante dispendiosas e apenas ceras de parafina de grau técnico devem ser utilizadas como PCM [11]. De acordo com [13], ao aumentar a cadeia de alcanos, a temperatura de fusão (T_{fus}) aumenta. Na tabela 2.2, encontram-se a T_{fus} e ΔH_{fus} em função do número de átomos que compõem a cadeia.

Tabela 2.2: Características de Ceras de Parafina Puras. (Adaptado de [13]).

Numero de átomos de carbono	T_{fus} (°C)	ΔH_{fus} (Kj/Kg)
15	10.0	205
16	16.7	237
17	21.7	213
18	28.0	244
19	32.0	222
20	36.7	246
21	40.2	200
22	44.0	249
23	47.5	232
24	50.6	255

2.3.1.2 Compostos sem Parafina

É dentro desta categoria que se encontra a maior quantidade de PCMs com as mais diversas propriedades físicas e químicas. Estes compostos, à semelhança dos compostos à base de parafina (em 2.3.1.1) são inflamáveis e possuem uma baixa condutividade

²Estado particular de um material que se encontra líquido enquanto a sua temperatura é inferior à temperatura de congelamento.

térmica, ver tabela 2.1 para mais detalhes. Estes compostos podem ser consequentemente divididos nas seguintes categorias:

- **Ácidos Gordos:** Possuem um elevado calor latente de fusão (ΔH_{fus}) comparável aos valores das ceras de parafina e apresentam um comportamento estável aquando da fusão e solidificação, estando também ausentes de super-arrefecimento. No entanto possuem a desvantagem de serem 2-2.5 vezes mais dispendiosos que as ceras de parafina de grau técnico e são ligeiramente corrosivos [14]. Na tabela 2.3, é possível observar T_{fus} e ΔH_{fus} em função do número de átomos que compõem a cadeia.

Tabela 2.3: Características de Ácidos Gordos. (Adaptado de [7].)

Nome Comum	Nº de átomos de C.	T_{fus} (°C)	ΔH_{fus} (Kj/Kg)
Ácido Caprílico	8	16.1	144.2
Ácido Cáprico	10	31.5	155.5
Ácido Dodecanoico	12	43.6	184.3
Ácido Mirístico	14	54.7	189.7
Ácido Palmítico	16	61.3	197.9
Ácido Esteárico	18	66.8	259.0

- **Glicóis:** ou glicóis de polietileno são solúveis em água e em compostos orgânicos [7]. Possuem também características desejáveis em PCMs como baixa pressão de vapor quando fundidos, são química e termicamente estáveis, não inflamáveis, biodegradáveis, não tóxicos e por último não são corrosivos [15]. Na tabela 2.4, é possível observar T_{fus} e ΔH_{fus} em função da massa molar (M) média.

Tabela 2.4: Características de Glicóis. (Adaptado de [15]).

Massa Molar média M ($g \cdot mol^{-1}$)	T_{fus} (°C)	ΔH_{fus} (Kj/Kg)
400	3.2	91.4
600	22.2	108.4
1000	32.0	149.5
1500	46.5	176.3
2000	51.0	181.4
3400	56.6	174.1
4000	59.7	189.7
6000	64.8	189.0
10000	66.0	189.6
20000	68.7	187.8

2.3.2 Materiais inorgânicos

PCMs Inorgânicos podem ser divididos em dois subgrupos: **metálicos** e **hidratos de sal**. Os PCMs metálicos são raramente utilizados graças ao seu preço e disponibilidade

reduzida [13].

2.3.2.1 Metálicos

Apesar de serem raramente utilizados na maioria das aplicações, existem condições e aplicações especiais onde se torna vantajoso o uso destes PCMs, nomeadamente em condições onde o espaço de trabalho é reduzido, (e.g. no campo da Astronáutica). Estes materiais são assim utilizados, pois possuem um calor latente de fusão por unidade de volume e condutividade térmica superior aos outros tipos de PCMs [14]. Na tabela 2.5, é possível ver uma lista de PCMs e as suas características.

Tabela 2.5: Características de metais. (Adaptado de [14]).

Material	T_{fus} (°C)	ΔH_{fus} (Kj/Kg)
Gálio	30	80.3
<i>Cerrolow</i> Eutético	58	90.8
Bi-Cd-In Eutético	61	25
<i>Cerrobend</i> Eutético	70	32.6
Bi-Pb-In Eutético	70	29

2.3.2.2 Hidratos de sal

É nesta categoria que se encontram os PCMs mais conhecidos e mais estudados. Estes compostos, caracterizados pela formula química $AB \cdot nH_2O$ (AB representa um sal inorgânico), são ligas de sais inorgânicos e água [13, 16]. Além das características descritas na tabela 2.1, estes materiais possuem também:

- Abundância de materiais diferentes no mercado;
- Baratos com factor custo beneficio elevado;
- Toxicidade ligeira.

Aquando da fusão existem três comportamentos que podem ser esperados por estes materiais: fusão congruente, fusão incongruente e fusão semi-congruente. Uma explicação destes tipos de comportamentos encontra-se na secção 2.3.

A tabela 2.6, resume as características dos hidratos de sal pertencentes à categoria dos materiais inorgânicos.

Tabela 2.6: Características Hidratos de Sal. (Adaptado de [14]).

Material	T_{fus} (°C)	ΔH_{fus} (Kj/Kg)
$CaCl_2 \cdot 12H_2O$	29.8	174
$LiNO_3 \cdot 2H_2O$	30.0	296
$LiNO_3 \cdot 3H_2O$	30.0	189
$KFe(SO_4)_2 \cdot 12H_2O$	33.0	173
$CoSO_4 \cdot 7H_2O$	40.7	270
$Ca(NO_3) \cdot 4H_2O$	47.0	153
$Ca(NO_3)_2 \cdot 3H_2O$	51.0	104
$NaOH \cdot H_2O$	64.3	273
$MgCl_2 \cdot 6H_2O$	117	167

2.3.3 Eutéticos

Um material eutético é uma composição de dois ou mais componentes. Sendo que, cada um destes materiais possui propriedades congruentes na fusão e solidificação, formando uma mistura de cristais no momento da cristalização [17].

Adicionalmente, estão ausentes de segregação de fase e possuem uma temperatura de fusão bastante regular. Sendo esta, um pouco inferior à temperatura de fusão dos materiais de que é constituído o PCM, a esta temperatura chama-se "temperatura eutética" [7, 14]. Na tabela 2.7, encontram-se alguns materiais eutéticos e as suas características.

Tabela 2.7: Características Eutéticos. (Adaptado de [14]).

Material	Composição (%)	T_{fus} (°C)	ΔH_{fus} (Kj/Kg)
Trietiloletano + Água + Ureia	38.5+31.5+30	13.4	160
$CaCl_2 + MgCl_2 \cdot 6H_2O$	50+50	25	95
$NH_2CONH_2 + NH_4NO_3$	53+47	46	95
$Mg(NO_3)_3 \cdot 6H_2O + MgCl_2 \cdot 6H_2O$	58.7+41.2	59	132
Naftalina + Ácido Benzoico	67.1+32.9	67	124
$LiNO_3 + NH_4NO_3 + NH_4Cl$	27+68+5	81.6	108

2.4 Métodos de incorporação de PCMs

A Incorporação de PCM consiste na inserção do mesmo em outros materiais ou meios recorrendo a um conjunto de métodos. De acordo com [18] os métodos de incorporação mais usuais são:

- **Incorporação Directa:** É o método mais simples no qual se insere o PCM directamente em outros materiais e.g. adicionar PCM directamente a gesso ou cimento num edifício. De acordo com [18] a temperatura de fusão pode ser alterada e é necessário ter em consideração a compatibilidade do PCM com os materiais em que este irá ser incorporado;

- **Imersão:** Ao contrário do método anterior, em que o PCM é incorporado no momento em que o composto final é criado, neste método, o PCM é absorvido pelo material base. Sendo assim, a incorporação é feita mergulhando o material num banho de PCM, fazendo com que este seja absorvida pelos poros do material por efeito de acção capilar³ [10]. É de notar que este método pode sofrer perdas de PCM a longo prazo;
- **Encapsulamento**, composto por duas técnicas, macro e micro encapsulamento:
 1. **Macro-encapsulamento:** consiste no encapsulamento de PCM em tubos, painéis ou esferas, de forma a evitar o contacto do PCM com o ambiente exterior, no caso de o recipiente ser rígido acrescenta durabilidade mecânica. Com este método, o problema das fugas é evitado e ao ser aplicado a estruturas estas são menos afectadas. Possui, no entanto, as desvantagens de condutividade térmica reduzida, tendência para solidificar nos cantos e ser difícil de se implementar. Na figura 2.5, é possível observar um painel flexível contendo PCM com o intuito de diminuir o consumo energético de sistemas Aquecimento Ventilação e Ar Condicionado (AVAC);



Figura 2.5: Pannel flexível contendo PCM.

2. **Micro-encapsulamento:** Consiste no encapsulamento de PCM num polímero ou membrana, com dimensões na ordem dos μm . Uma das grandes vantagens desta tecnologia é o facto de aumentar a relação superfície-volume, efectivamente melhorando a transferência de calor por unidade de volume [8]. Esta tecnologia permite, facilidade de manuseamento, ausência de fugas e manutenção do PCM [13]. Na figura 2.6, está representado uma micro cápsula de PCM *Micronal*TM, com um diâmetro de aproximadamente $5\mu m$, onde o PCM utilizado é cera.

³Ocorre quando a força de adesão é superior à de coesão, fazendo com que um líquido seja atraído por um sólido.

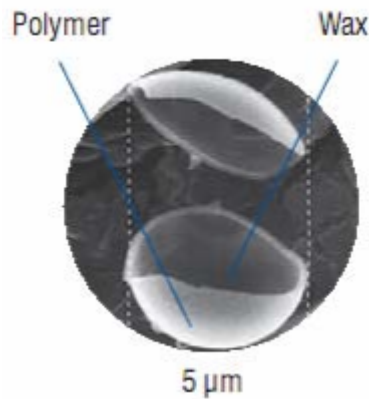


Figura 2.6: Micro cápsula PCM. (Adaptado de [13]).

2.4.1 Aplicações

Em adição às propriedades descritas nas secções anteriores e de acordo com [19], existem duas características chave que fazem com que a indústria procure a implementação de PCM nos seus sistemas. Estas características são: a capacidade de estabilizar termicamente um sistema, reduzindo assim variações de temperatura do sistema e a capacidade de armazenar energia calorífica. Posto isto, destacam-se assim as seguintes aplicações de PCM na indústria:

- Aplicações na **indústria culinária**, para transporte de alimentos a temperaturas constantes ou para utilização em electrodomésticos industriais (e.g. fornos, frigoríficos e congeladores);
- Aplicações **médicas**, maioritariamente para transporte de sangue ou órgãos, no entanto, existem também aplicações em colchões para recém-nascidos para o tratamento de hipoxia;
- Aplicações **electrónicas**, utilizado normalmente para dissipação de calor;
- Aplicações **arquitectónicas**, armazenamento de energia térmica durante as horas de maior calor;
- Aplicações de Aquecimento Ventilação e Ar Condicionado (**AVAC**).

Como a maioria destes sistemas têm como função o aquecimento de um material ou de um fluido, é possível determinar que o seu funcionamento é composto por um elemento de aquecimento e um elemento de controlo de temperatura à excepção de casos muito específicos, os componentes utilizados são uma resistência e um termostato respectivamente.

Ao instalar estes materiais, PCMs, num sistema como o descrito anteriormente, é necessário obter dados que comprovem que o sistema, após essa mesma instalação, irá beneficiar das características estudadas. Sendo assim, é necessária uma solução de monitorização, com a função de analisar parâmetros como a potência eléctrica consumida ao longo do seu funcionamento e as várias temperaturas presentes na montagem. Deste modo, irá ser estudado na secção seguinte o conceito de *Internet of Things* (IoT) e as suas possíveis aplicações para a monitorização integrada de sistemas.

2.5 *Internet of Things* para a Monitorização Integrada de Sistemas

Ao longo dos anos houve uma acentuada evolução no que toca à tecnologia computacional e às suas aplicações. Começando em 1999, o termo IoT foi introduzido por Kevin Ashton, termo este que ficou difundido por todo o mundo, graças ao trabalho produzido no *Auto-Id Centre*, grupo de investigação no campo da identificação por *Radio-Frequency Identification* (RFID) [20]. Com a evolução da tecnologia e do conceito de IoT, o sistema RFID manteve-se na vanguarda do conceito IoT orientado as “coisas”, devendo-se isto à maturidade do mesmo, ao baixo custo e ao forte apoio da indústria. Contudo, também se constatou o crescimento várias de tecnologias como: *Near Field Communication* (NFC), *Wireless Sensor and Actuator Networks* (WSAN) e *Wireless Identification and Sensing Platforms* (WISP). Sendo que, neste caso se visava o conceito de IoT numa vertente orientada à “internet” [21].

Atualmente a tecnologia IoT encontra-se bem estabelecida, sendo possível aplicar a mesma a um grande quantidade de áreas, desde aplicações pessoais a industriais. Devido a este crescimento e ao facto de esta se encontrar presente em diversas aplicações, o foco de interesse apresenta-se maioritariamente sobre as tecnologias de privacidade e segurança[22].

Posto isto, destacam-se assim as seguintes aplicações de IoT:

- Comunicação entre máquinas (**indústria 4.0**);
- Comunicação entre máquinas e infraestruturas (**indústria 4.0**);
- **Smart applications** e.g. Cidades, Edifícios, Habitação, Agricultura, carros, estacionamento, entre outros;
- Monitorização, controlo industrial e Controlo de qualidade;
- Segurança remota;

O conceito de IoT refere-se à interconexão de dispositivos com a *Internet*, ou seja, é uma rede hierarquizada de dispositivos, com a capacidade de adquirir, processar e

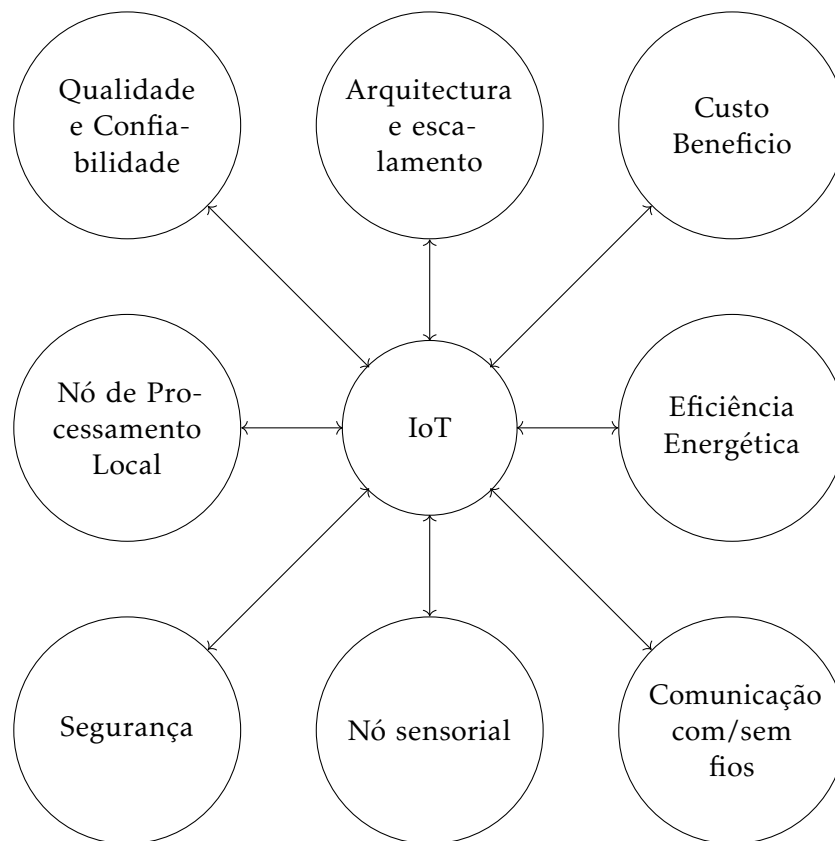


Figura 2.7: Conceitos essenciais para IoT.

transmitir dados. Na figura 2.7, está presente um diagrama com diferentes conceitos, que podem ou devem, estar presentes numa aplicação IoT.

De modo a obter um nó IoT operacional, é necessário projectar uma arquitectura que se adequa ao projecto em questão. Para tal, é necessário a utilização de diversos módulos, cada um com uma função específica, para que o objectivo final seja alcançado. De entre esse módulos, estão alguns como:

- MCU ou micro-controlador;
- Módulos de comunicação;
- Módulos de Sensores;
- *Analog to Digital Converter (ADC)*.

2.5.1 Micro-Controladores

O MCU ou micro-controlador, tem como função adquirir, processar, armazenar e exportar os dados provenientes dos sensores que são utilizados em simultâneo com o micro-controlador.

Neste capítulo, devido à enorme variedade de micro-controladores, ir-se-á apenas estudar os que mais se têm destacado no mercado durante os últimos anos. Visto que, estes mesmos micro-controladores vêm acompanhados de plataformas de desenvolvimento, não se irá utilizar o nome dado ao micro-controlador, mas sim, o nome da plataforma em si. De entre os micro-controladores existentes, os que mais se destacaram no mercado são:

- Arduino;
- ESP;
- Psoc.
- Raspberry Pi;

2.5.1.1 *Arduino*

O *Arduino*, é uma plataforma de desenvolvimento *open-source*⁴, criada em 2005 por Hernando Barragán, com o principal objectivo de ser fácil de utilizar, em termos de *hardware* e *software*. À semelhança de outros micro-controladores, o *Arduino* já sofreu também ao longo dos anos várias iterações, possuindo até este momento inúmeros modelos, cada um com diferentes pontos fortes e fracos (figura 2.8).



Figura 2.8: Produtos *Arduino*.

Esta plataforma de desenvolvimento possui várias versões, visando diferentes tipos de aplicações, sendo a mais conhecida, o *Arduino UNO*. Esta plataforma encontra-se

⁴Modelo de desenvolvimento, que promove licenciamento livre para o design ou esquematização de um produto e a sua subsequente redistribuição universal, com a possibilidade de livre consulta, examinação ou modificação do mesmo, sem que seja necessário pagar uma licença comercial.

equipada com um micro-controlador de 8bits, funcionando a 16MHz, possuindo pontos fortes como a componente *open-source*, a simplicidade e facilidade da linguagem de programação, o seu *Integrated Development Environment* (IDE), a interoperabilidade entre plataformas e o seu preço. Apesar do *Arduino uno* ser relativamente limitado em termos de velocidade de *clock* e de possuir uma quantidade reduzida de memória, existem outros modelos superiores, e.g. o *Arduino DUE* encontra-se equipado com um processador ARM de 32bits, funcionando a 84MHz e possui uma quantidade de memória superior.

2.5.1.2 ESP32

O ESP32 (figura 2.9) é um módulo de processamento pertencente à família ESP, lançada em 2014 pela *Espressif Systems*, este apresenta apenas a opção de se utilizar como um todo, não existindo a possibilidade de se utilizar apenas o micro-controlador de forma isolada [23, 24].



Figura 2.9: ESP32.

Este módulo encontra-se bastante completo e com um elevado poder de processamento, devido ao seu processador com dois núcleos podendo atingir frequências de *clock* de 240MHz, antenas *Wi-Fi* e *Bluetooth V4.2* com suporte para *Bluetooth Low Energy* (BLE) e dois ADC de 12bits do tipo SAR. Ao implementar este módulo ESP32 é necessário ter em considerações as limitações da tensão de alimentação, visto que este suporta apenas uma alimentação e nível lógico de 3,3V, dificultando assim a implementação com módulos que utilizem um nível lógico de 5V.

2.5.1.3 PSoC 5Lp

O *Programmable System-on-Chip* (PSoC) presente na figura 2.10, foi criado nos Estados Unidos da América pela *Cypress Semiconductor* em 2002. Com a ajuda de um desafio, *PSoC Design Challenge*, patrocinado na revista *Circuit Cellar*, começou a ser comercializado em quantidades elevadas.



Figura 2.10: PSoC 5Lp.

Na figura 2.11 é possível observar o diagrama de blocos do PSoC 5LP, mais especificamente da família CY8C58LP. Onde é possível observar que o MCU se encontra dividido em quatro secções contendo grupos específicos de blocos que se encontram inter-conectados por um barramento de sistema. Estas quatro secções são:

- **Sub-sistema do CPU:** Esta secção encontra-se construída em torno de um processador ARM Cortex-M3 de 32bits, com uma frequência até *clock* até 80MHz. Nesta mesma secção encontram-se também os blocos responsáveis pelo controlo dos vários tipos de memória presentes no MCU e pela programação e depuração do mesmo;
- **Recursos gerais do sistema:** Esta secção engloba dois blocos distintos, o primeiro é responsável pela geração das diferentes frequências de *clock* do sistema, estando o mesmo equipado com um gerador variável de *clock* interno projectado para alta estabilidade e precisão. O segundo é responsável pela regulação da tensão de alimentação, podendo ser alimentado com tensões desde 0,5V até 5V, devido à utilização conversor de tensão síncrono em modo boost;
- **Sistema digital:** Tem como função a conexão de sinais digitais provenientes de quaisquer tipo de periféricos a qualquer pino à escolha. Adicionalmente este bloco providencia um grande quantidade de periféricos digitais através de *Universal Digital Blocks* (UDB), 4 temporizadores de 16bits, funções *Pulse Width Modulation* (PWM) e comunicações digitais como *Inter-Integrated Circuit* (I2C) e *Universal Serial Bus* (USB);
- **Sistema analógico:** Tem como função o roteamento de sinais analógicos para interior e exterior do dispositivo através de um barramento analógico. Adicionalmente, contem todos os periféricos analógicos do MCU, sendo os mais relevantes um ADC sigma-delta configurável até 20bits e 2 do tipo SAR com uma capacidade de 12bits;

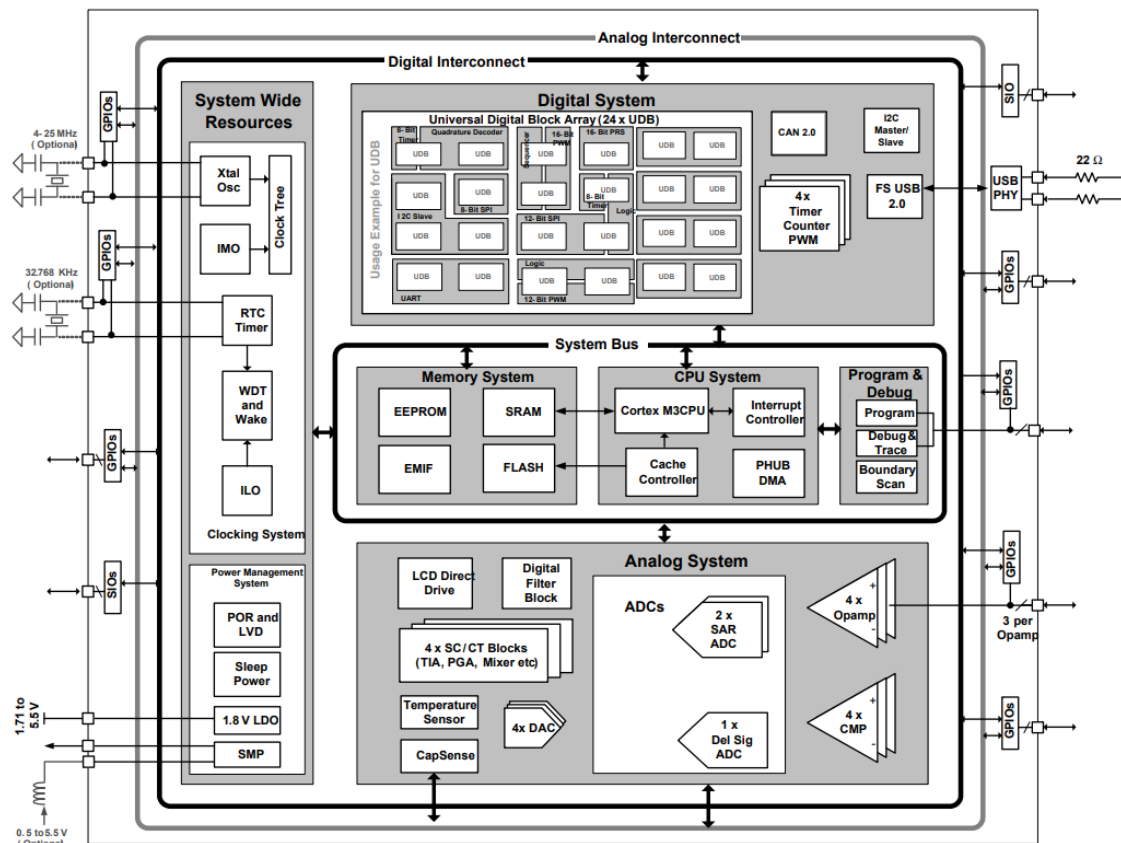


Figura 2.11: Diagrama de blocos PSoC 5LP, família CY8C58LP. (Adaptado de [25]).

Em suma, o PSoC está munido de um processador 32bits ARM Cortex-M3 com uma frequência até 80MHz, com imensos componentes e recursos à sua disposição, desde *Digital to Analog Converter* (DAC), filtros, *Operational Amplifier* (OpAmp), *Real Time Clock* (RTC), Interfaces de comunicação, entre outros. Por fim, à semelhança do Arduino, o PSoC possui um IDE próprio com o nome de PSoC Creator onde é possível tomar partido de todas as funcionalidades oferecidas pelo PSoC.

2.5.1.4 Raspberry Pi

O *Raspberry Pi* (figura 2.12), apesar não ser um micro-controlador encontra-se nesta subsecção, visto que, devido ao seu tamanho e preço é comumente utilizado para a realização de tarefas mais exigentes, onde não é possível a utilização de um MCU convencional ou a implementação de um processador de alta performance. O *Raspberry Pi* é um pequeno computador do tamanho de um cartão de crédito, lançado em 2012 no Reino Unido pela fundação *Raspberry Pi*. Ao longo dos anos tem havido inúmeras iterações deste pequeno computador, sendo que a primeira possuía um processador com um único núcleo a funcionar a 700MHz com apenas 256MB de *Random Access Memory* (RAM), o último modelo, no entanto, já possui um processador com quatro núcleos a 1.4GHz e 1GB RAM.

Este computador, está munido de várias entradas de periféricos e I/O, sendo capaz

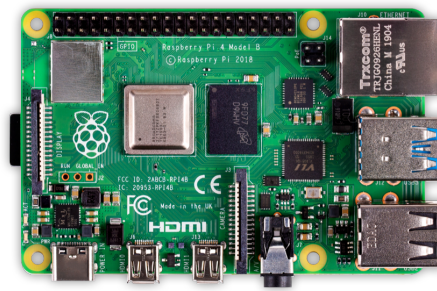


Figura 2.12: Raspberry Pi modelo 4 B.

de ser utilizado com vários sistemas operativos. Podendo também, ser utilizado para aplicações que necessitem de um poder de processamento elevado. No entanto, não possui ADC, nem um RTC fazendo com que aplicações que necessitem de um controlo em tempo real, ou de análise de dados provenientes de sensores analógicos necessitem de módulos complementares.

2.5.1.5 Análise comparativa dos micro-controladores

Como demonstrado nos capítulos supracitados, existe uma quantidade bastante abrangente de micro-controladores disponíveis no mercado. Sendo assim, esta secção tem como objectivo expor os pontos forte de cada micro-controlador presente na tabela 2.8. É de notar, que os factores de comparação, devem ser adaptados à aplicação que se está a implementar, e não de uma maneira generalista. Deste modo, foram escolhidos os seguintes pontos: Sistema Operativo, Linguagem de programação, ADCs, RTCs, Wi-Fi, *Bluetooth*, potência, dimensão e preço.

Tabela 2.8: Comparação dos Micro-controladores.

	Raspberry Pi 3B+[26]	Arduino UNO[27]	ESP32[28]	PSoC 5Lp[25]
Processador	BCM2837B0	ATmega238p	Xtensa LX6	ARM CortexM3
Frequência	1.4GHz	16MHz	240MHz	80MHz
Tensão	5V	5-12V	3,3V	0.5-5V
Sistema Operativo	Linux	N/A	N/A	N/A
Linguagem de Programação	Qualquer	C++ (Arduino IDE)	Micro Python C	C (Psoc Creator IDE)
ADC	Não	10bits SAR	12bit SAR	20bit $\Sigma\Delta$ 12bit SAR
INL*	N/A	± 1.5	± 12	± 32 & ± 2
DNL*	N/A	± 0.7	± 7	± 1 & ± 2
RTC	Não	Sim	Sim	Sim
Wi-Fi	Sim	Não	Sim	Não
Bluetooth	Não	Não	Sim	Não
Potência (mW)	≈ 4000	≈ 230	≈ 50	≈ 77
Dimensões (mm)	84×56	65×50	50×30	110×24
Preço (€)	$\approx 40€$	$\approx 30€$	$\approx 10€$	$\approx 14€$

Podem existir discrepâncias consoante os modelos e data de fabrico.

*Integral Nonlinearity (INL); *Differential Nonlinearity (DNL).

2.5.2 Analog to Digital Converters

Nesta secção, ir-se-ão estudar os componentes essenciais para a obtenção de sinais analógicos e a sua conversão para um sinal digital. Como este componente está responsável pela conversão do sinal analógico, a sua exactidão e precisão necessitam de ser elevadas, pois algum erro gerado nesta fase, ir-se-à propagar ao longo de todo o processo. Posto isto, irão ser estudados os dois tipos de ADCs mais comuns, sendo que, estes estão também presentes nos micro-controladores apresentados na tabela 2.8.

2.5.2.1 Successive Approximation Register (SAR) ADC

Successive Approximation Register (SAR), ou em português registo de aproximações sucessivas, é uma classe de ADC, que como o nome indica, necessita de alguns ciclos de comparação e aproximação para que efectue a conversão do sinal recebido. Sendo assim, estes ADCs possuem tempos de operação reduzidos, fazendo com que sejam um dos tipos de ADC mais utilizados em aplicações de baixo consumo e velocidade [29]. Atendendo aos micro-controladores analisados anteriormente, este ADC encontra-se empregue em todos eles, à excepção do *Raspberry Pi*.

Na seguinte figura 2.13, é possível observar um diagrama simplificado do funcionamento do ADC. Adicionalmente, o bloco *Sample and Hold (S&H)* é responsável pela amostragem do sinal a converter, e o DAC encontra-se responsável por gerar a tensão de comparação.

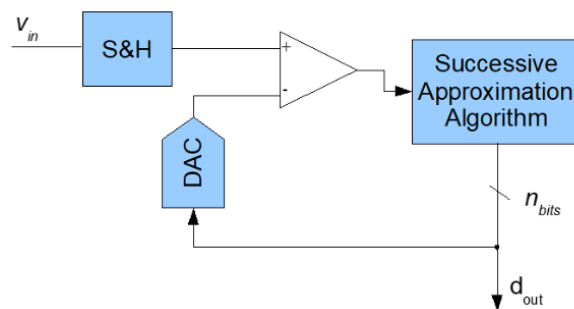


Figura 2.13: Funcionamento SAR ADC.

Este ADC, de modo a efectuar a comparação do sinal recebido, necessita de um algoritmo adicional, *SAR Logic* ou *Successive approximation algorithm*, onde são controladas as comparações entre o sinal amostrado e as tensões de referencia presentes no ADC. Na seguinte figura 2.14, é possível observar o fluxograma representativo do algoritmo.

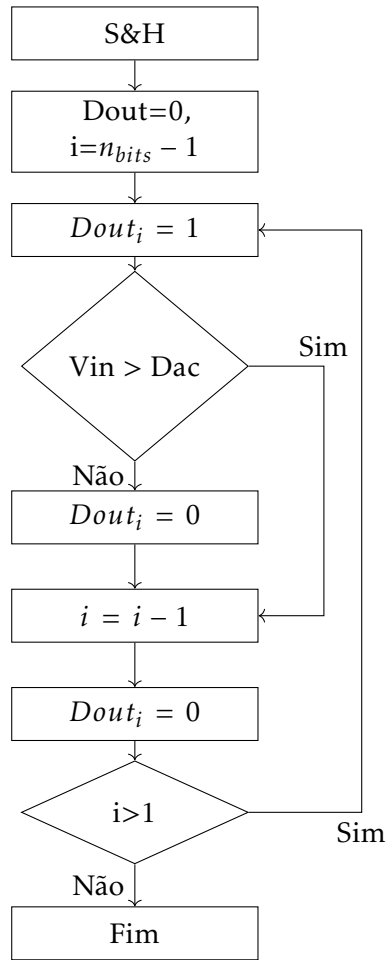
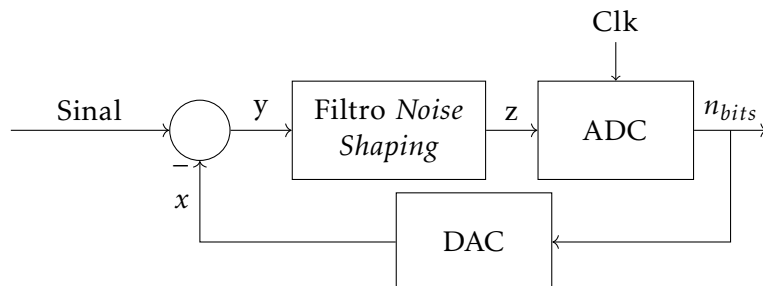


Figura 2.14: Fluxograma SAR Logic.

2.5.2.2 Sigma-Delta ADC

Este tipo de ADC apoia-se no princípio da sobre-amostragem, figura 2.15, de maneira a dividir o ruído proveniente da quantização de um sinal numa maior largura de banda. Posto isto, é possível aumentar a relação de sinal ruído, fazendo efectivamente com que a resolução do DAC aumente [30].

Figura 2.15: Diagrama simplificado ADC Sigma Delta e *noise shaping*.

Ao utilizar um filtro de *noise shaping* é possível deslocar, mas não eliminar, o ruído restante para fora da banda do sinal que se quer converter. Na seguinte figura 2.16 é possível observar o efeito do filtro de *noise shaping*.

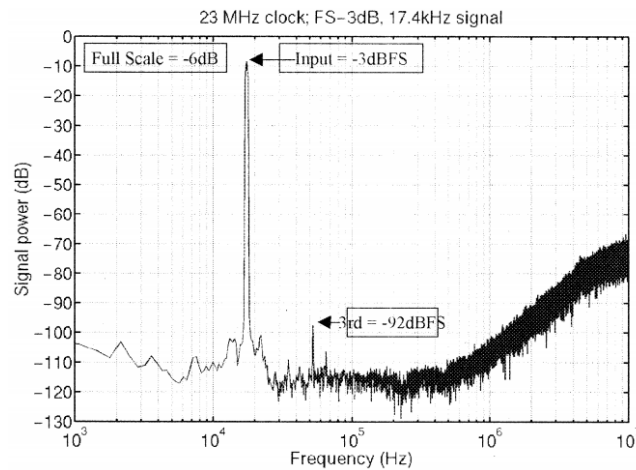


Figura 2.16: Análise espectral de ADC Sigma-Delta. (Adaptado de [30]).

Sendo assim, este ADC devido ao seu funcionamento, oferece uma maior resolução e uma maior mitigação de ruído em relação ao ADC do tipo SAR, no entanto, estas características são restringidas pela sua velocidade, limitando assim as suas aplicações e fazendo com que não seja aconselhado a utilização do mesmo para gamas superiores às das frequências sonoras. Adicionalmente, este ADC encontra-se utilizado por apenas um dos micro-controladores analisados na secção anterior, sendo este o PSoC 5LP.

2.5.3 Módulos de sensores

Os módulos de sensores são possivelmente o elemento mais básico presente num nó IoT. No entanto, são também um dos mais importantes, pois sem estes torna-se impossível captar dados de um dado ambiente. De acordo com dados presentes nos capítulos, é possível determinar que poderão ser necessários os seguintes sensores:

- Sensor de **temperatura**;
- Sensor de **nível**;
- Sensor de **energia eléctrica**.

2.5.3.1 Sensores de temperatura

A medição da temperatura em vários pontos de um sistema é crucial para o estudo dos benefícios que o PCM poderá trazer. Dadas as condições em que o sistema poderá estar, será necessário que este seja compatível em uma vasta gama de temperaturas, e que seja de

alguma maneira à prova de água. Adicionalmente, será benéfico se estes sensores possuírem um, ou mais, protocolos de comunicação, sendo estes *one-wire*, *I2C*, *Serial Peripheral Interface* (SPI) ou qualquer forma de comunicação wireless.

De entre os diversos tipos de sensores de temperatura é possível destacar três, sendo estes: termistor, termopar e o *Resistance Temperature Detector* (RTD) ou em português detector de temperatura por resistência.

O **termistor** é um dispositivo semi-condutor com uma ligação de 2 fios, estes sensores têm como funcionamento básico a variação da sua resistência interna em função da temperatura. Dentro desta categoria de sensores existem duas classes, os NTC e os PTC. Os sensores NTC estão associados à diminuição da sua resistência em função do aumento da temperatura e os PTC estão associados ao aumento da resistência em função do aumento da temperatura. Por fim, estes sensores são bastante utilizados devido à sua elevada estabilidade e precisão associadas às suas dimensões e preços reduzidos.

De seguida tem-se os sensores de temperatura do tipo **termopar**, estes são compostos pela junção de dois metais diferentes, que devido ao efeito de *Seebeck* geram uma diferença de potencial entre os mesmos. É possível, com este tipo de sensores, medir uma variada gama de temperaturas ao alterar o tipo de junção, ou seja, alterar os tipos de metais que constituem a mesma.

Por último, tem-se os sensores de temperatura **RTD** que à semelhança do termistor variam a sua resistência em função da variação de temperatura. No entanto, estes não são um semi-condutor, mas sim uma termo-resistência composta normalmente por platina ou níquel. Um exemplo bastante conhecido de um sensor RTD é a famosa PT-100, sendo esta uma termo-resistência de platina que apresenta uma resistência de 100Ω a 0°C .

Posto isto, foi possível encontrar dados sobre dois sensores, presentes na tabela 2.9, que se adequam às especificações, sendo estas: a capacidade de medir uma vasta gama de temperaturas, possuir algum tipo de protocolo de comunicação e a capacidade de ser à prova de água.

Tabela 2.9: Sensores de temperatura.

Sensor	DS18B20 [31]	MNS-9-W1-TS-WT-L03 [32]
Gama ($^{\circ}\text{C}$)	-55°C a 125°C	-40°C a 125°C
Comunicação	<i>1-wire</i>	Rádio frequência
À prova de água	Sim	Sim
Preço (€)	$\approx 8\text{€}$	$\approx 60\text{€}$

2.5.3.2 Sensores de nível

Como o nome indica, este sensor tem como função medir o nível de um certo líquido, garantindo assim que o sistema poderá trabalhar com a quantidade adequada de fluido. Estes sensores podem ser divididos em duas classes: medição contínua ou medição discreta. Medição contínua indica continuamente o nível do fluido enquanto este sobe e

desce, por outro lado, medição discreta indica quando um fluido está em certos pontos definidos (e.g. máximo, metade e mínimo). Existem vários tipos de sensores disponíveis, sendo estes os mais comuns:

- Ópticos;
- Ultra sónicos;
- Flutuantes;
- Capacitivos;
- Resistivos.

Devido à construção do equipamento descrito no capítulo 3.1, será difícil de equipar este tipo de sensor, no entanto, foi possível encontrar alguns dados sobre um sensor que poderá ser utilizado. Este sensor é o NC57, produzido por *FISCHER Messtechnik* [33], custando aproximadamente 200€. Este é um sensor discreto, fazendo apenas medidas em pontos específicos.

2.5.3.3 Sensor de energia eléctrica

A medição de energia eléctrica é também um factor bastante importante, pois é necessário verificar se a eficiência do sistema melhora com a introdução de PCM. Para tal, é necessário um módulo que tenha a capacidade de medir no mínimo potência activa (P), factor de potência, frequência e valores RMS de tensão e corrente. Após alguma pesquisa, foi possível encontrar os módulos presentes na tabela 2.10.

Tabela 2.10: Módulos medição de energia eléctrica.

Sensor	MSP430AFE2x3 [34]	78M6613 [35]	ACS71020 [36]
Fabricante	<i>Texas Instruments</i>	<i>Maxim Integrated</i>	<i>Allegro Micro</i>
Tensão e corrente RMS	Sim	Sim	Sim
Potência(S,Q,P)*	Sim	Sim	Sim
Fator de potência	Sim	Sim	Sim
Frequência	Sim	Sim	Sim
Comunicação	UART/SPI	UART	SPI/I2C
Preço (€)	≈5€	N/a	≈5€

Podem ser necessários componentes adicionais.

*Potência(S,Q,P): Aparente; Reativa e Ativa respetivamente.

PROJETO DA ARQUITETURA E IMPLEMENTAÇÃO DO SISTEMA DE MONITORIZAÇÃO PARA PCM

Neste capítulo, primeiramente, é abordado o equipamento que se pretende equipar com PCM e consequentemente monitorizar o seu funcionamento. Seguidamente, é abordada a arquitetura do sistema, onde são explicadas todas as características, especificações e decisões tomadas referentes à mesma, visto ser fulcral para a implementação de qualquer projeto. Na secção referente ao módulo de sensores, são apresentados os conceitos chave e parâmetros de funcionamento deste módulo, bem como os seus constituintes como o módulo de monitorização de potência e temperatura onde serão identificados os recursos que estão ao ser dispor e o dimensionamento necessário para o bom funcionamento dos mesmos.

Na secção relativa ao módulo de aquisição de dados, são apresentadas as características chave do módulo, seguida de uma explicação, de forma generalizada, do seu funcionamento. Nas restantes secções, são abordados genericamente, os pontos fundamentais e as escolhas realizadas referentes ao módulo de comunicação e à escolha da plataforma IoT, bem como, uma breve explicação do funcionamento dos mesmos.

3.1 Equipamento a utilizar

De maneira a obter dados experimentais válidos é necessário criar um cenário de testes, onde seja possível aplicar os conceitos estudados nos capítulos anteriores. Para tal, ir-se-á utilizar um equipamento industrial pertence à área da restauração, pois este, possui um funcionamento adequado ao teste de PCMs.

Este equipamento trata-se de uma chocolateira industrial, Figura 3.1, da marca *Campeona* [37] que faz uso de um de painel de aquecimento (método indireto) para confeccionar o chocolate, sendo bastante fácil a introdução de PCM no mesmo. O painel de aquecimento

CAPÍTULO 3. PROJETO DA ARQUITETURA E IMPLEMENTAÇÃO DO SISTEMA DE MONITORIZAÇÃO PARA PCM

recorre à utilização de um líquido, nomeadamente água, para se aquecer de forma lenta, indireta e uniforme uma dada substância até uma temperatura definida.



Figura 3.1: Equipamento a utilizar - Chocolateira industrial.

Após uma análise do funcionamento deste equipamento, foi possível determinar que o mesmo é composto por:

1. **Termostato**, responsável pelo controlo de temperatura;
2. **Elemento elétrico** de aquecimento, responsável pelo aquecimento da água utilizada no painel de aquecimento;
3. **Cuba** em contacto com água, de modo a aquecer a substância.

De maneira a estudar o efeito que os PCM poderão ter na eficiência energética deste sistema, pretende-se introduzir sensores de temperatura no compartimento onde estará a água utilizada no painel de aquecimento, na cuba de aquecimento, no exterior do equipamento e monitorizar a potência elétrica que este sistema consome. Após completar os testes de controlo, pretende-se substituir a água utilizada no painel de aquecimento por um PCM e proceder a mais testes. No diagrama 3.2, é possível observar, com recurso a um diagrama de blocos o funcionamento do equipamento que se pretende utilizar

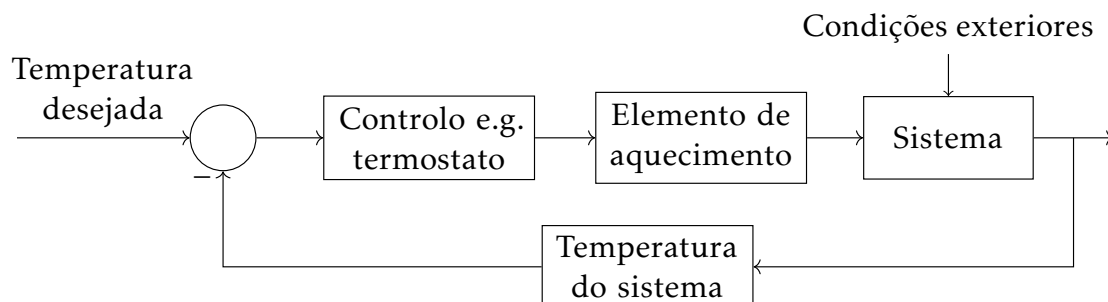


Figura 3.2: Diagrama de blocos do equipamento

3.2 Arquitetura do Sistema

Nesta dissertação é proposto o desenvolvimento e implementação de um nó IoT multisen-
sor, mais concretamente de um sistema com a capacidade de recolher, processar, exibir
e armazenar dados provenientes de diversas fontes em tempo real para uma plataforma
IoT. Apesar de se conhecer o equipamento a monitorizar anteriormente ao desenvolvi-
mento desta arquitetura, pretende-se que a mesma seja de fácil aplicação a um diverso
conjunto de máquinas e equipamentos, posto isto, é necessário criar uma arquitetura que
seja modular e reconfigurável.

Para cumprir as especificações descritas é necessário dividir a arquitetura em secções
cada uma com uma função específica. Ao fazer esta divisão será fácil garantir a modu-
laridade e a configurabilidade do sistema, restando assim a escolha das funções destas
diferentes secções.

Para este sistema em concreto onde se pretende recolher, processar, enviar e armazenar
dados serão necessárias quatro secções (figura 3.3), sendo estas, o módulo de aquisição de
dados, o módulos de sensores, o módulo de comunicação e por fim o módulo IoT que irá
representar a plataforma IoT e todos os seus serviços.

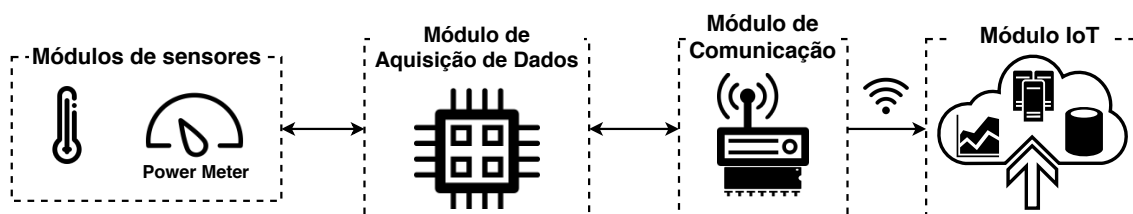


Figura 3.3: Arquitetura do sistema.

Como módulo de mais baixo nível temos o módulo de sensores, onde é feita a interface
com o equipamento a monitorizar e a recolha de dados físicos para o meio digital. Este
será o modulo que mais alterações vai sofrer conforme o equipamento que se pretende
monitorizar, visto que, cada equipamento terá um grupo de características e pontos críti-
cos diferentes que se pretendem observar, fazendo assim com que os sensores necessários
variem consideravelmente. Em seguida, temos o módulo de aquisição de dados, onde é
feita a interface e recolha dos dados provenientes dos sensores presentes no módulo des-
crito anteriormente, caso seja necessário algum processamento de dados o MCU presente
neste módulo estará encarregue dessa tarefa.

Ao fazer a divisão desta maneira será também possível agrupar os módulos de sen-
sores ao modulo de aquisição de dados numa camada superior, criando assim, o modulo
de monitorização. Com este modulo é então possível fazer um sistema altamente versá-
til, onde é possível recolher e transmitir dados de inúmeros módulos de monitorização
em simultâneo. Adicionalmente, ao agrupar o módulo de monitorização ao modulo de
comunicação obtém-se então o Nó IoT como se pode ver na figura 3.4.

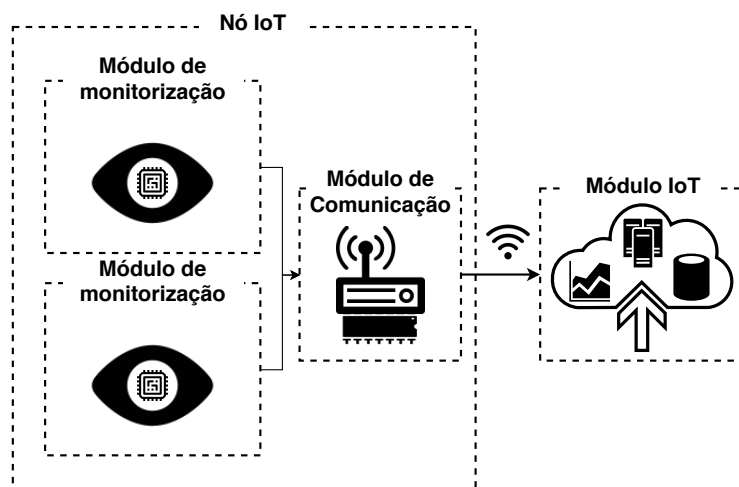


Figura 3.4: Arquitetura do sistema, agrupamento de módulos.

Por último, o módulo IoT que representa a plataforma e os seus serviços tem a capacidade de receber dados de inúmeros nós IoT, fazendo então com que seja fácil implementar um sistema sobre uma área de grandes dimensões. No entanto, ao tentar implementar este sistema sobre uma grande área existe um problema que se torna evidente, sendo este, a comunicação entre o nó IoT e plataforma IoT. Posto isto, ao equipar o nó IoT com módulos de comunicação com diferentes métodos de comunicação, com ou sem fios, é possível colmatar esta falha fazendo assim com que o sistema seja facilmente aplicável a virtualmente qualquer ambiente. Na seguinte figura 3.5, é possível observar um exemplo onde três nós IoT estabelecem uma conexão com a plataforma utilizando diferentes tipos de comunicação, neste caso, Wi-Fi, Ethernet e por último 5G.

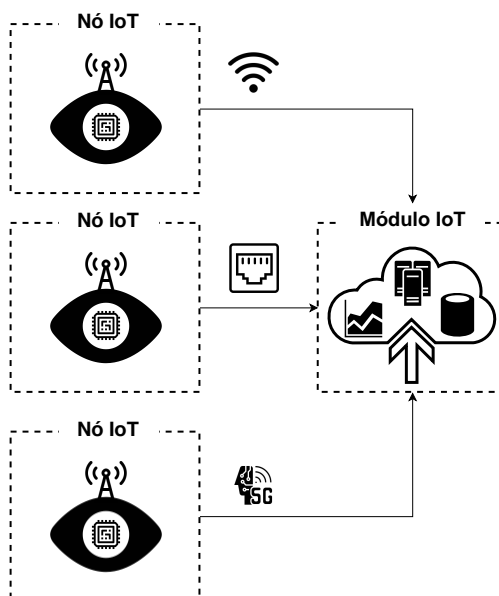


Figura 3.5: Arquitetura do sistema diferentes comunicações.

3.3 Módulo de sensores

Como um dos elementos mais simples deste sistema temos o sensor. O sensor tem como seu funcionamento básico a produção de um sinal de uma maneira específica, normalmente em tensão, correspondente a um estímulo físico, químico ou biológico. Este sinal é então processado de maneira a corresponder a uma grandeza física para fins de medição e ou de monitorização. De acordo com o sistema descrito em 3.1, as principais grandezas que se pretendem monitorizar são a temperatura, seja esta, ambiente ou intrínseca ao funcionamento do equipamento e o consumo energético do mesmo.

3.3.1 Medição de temperatura

Para efetuar medições de temperatura foi selecionado o sensor digital DS18B20 descrito na subsecção 2.5.3.1, pois este suporta uma elevada gama de temperaturas e tem a capacidade de enviar e receber informação proveniente do microcontrolador através de comunicação *One-Wire*.

Este sensor além de possuir as características acima descritas fornece uma leitura de temperatura entre *9bits* a *12bits*, sendo esta programável de acordo com os requisitos necessários, é de notar que com o aumento da resolução de medida advém também um maior tempo de conversão AD, chegando a *750ms* na resolução mais elevada. Possui também funções de alarme de excesso e/ou insuficiência de temperatura com recurso a memória não volátil programável pelo utilizador, um numero de série de *64bits* fazendo com que se possam ter múltiplos dispositivos no mesmo barramento de dados *One-wire* e por último a capacidade de funcionar com fonte de alimentação ou em modo parasita¹ [31].

3.3.1.1 Implementação

Atendendo ao facto de que para os sistemas a monitorizar a não existe a necessidade de ter um numero relativamente elevado de sensores, decidiu-se então, utilizar os sensores em modo "*single-drop*", sendo que este modo consiste em ter uma linha de dados para cada sensor. Na figura 3.6 é possível observar o esquema de ligação entre o sensor DS18B20 e um MCU, utilizando o modo "*single-drop*" com fonte de alimentação.

Posto isto, deixa de ser necessário a utilização de um algoritmo de pesquisa e armazenamento de números de serie presentes no barramento de dados *One-wire*, pois este sensor tem a possibilidade de receber comandos que não sejam endereçados a um sensor específico, utilizando o comando "*Skip rom - (CC)_h*" presente no *datasheet* [31], seguido do comando que se pretende utilizar, e.g. a ordem de conversão de temperatura.

¹Modo onde o sensor deriva a própria alimentação da linha de comunicação, eliminando assim a necessidade de uma fonte de alimentação externa.

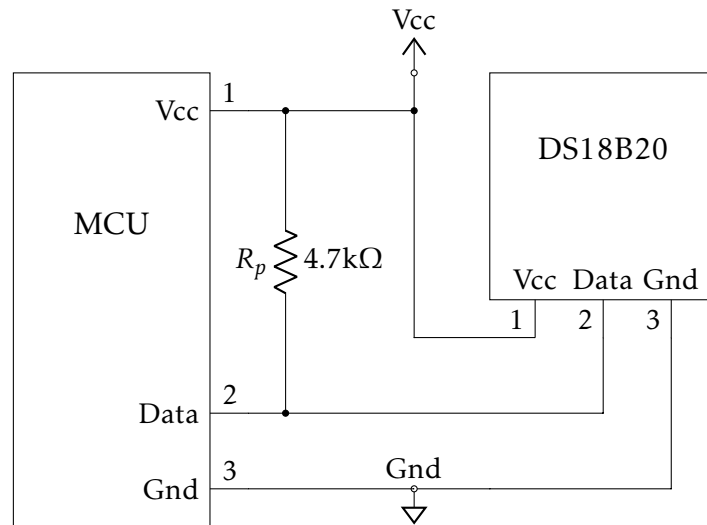


Figura 3.6: Ligação entre o sensor e o Microcontrolador em modo *Single-drop*.

Após definidas as condições de funcionamento resta então fazer a implementação dos sensores no MCU escolhido. Visto que o MCU é um PSoC 5Lp, descrito em 2.5.1.3, ir-se-á utilizar o *IDE PSoC Creator* onde é possível importar blocos com funções pré-programadas. Sendo que o comportamento destes blocos depois de ativos se assemelha à utilização de uma biblioteca de programação. Na figura 3.7 é possível observar um bloco DS18B20 com quatro terminais, cada um associado a um sensor de temperatura, com a função de realizar a interface com os respetivos sensores através de *One-wire*.

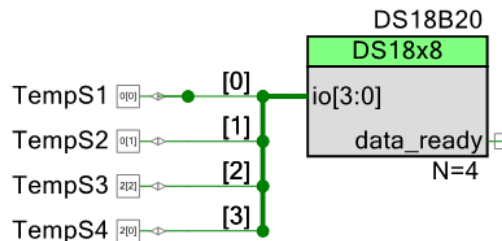


Figura 3.7: Implementação de sensores DS18B20 com recurso a um bloco de interface pré-programado.

Atendendo ao *datasheet* [38] do bloco utilizado e ao respetivo fluxograma representado na figura 3.8 é possível ver que o seu funcionamento é composto por 4 fases.

Na fase inicial o bloco encontra-se preparado. Caso se pretenda fazer uma medição de temperatura é efetuado um pedido de dados, após este pedido ocorre um período de espera de $750ms$ para que os sensores acabem a conversão de temperatura, é de notar que este período de espera é não bloqueante fazendo com que o MCU continue com o seu funcionamento. Decorridos os $750ms$ é ativada uma *flag* com o nome de "*_DataReady*" sinalizando que os dados estão preparados para serem recolhidos.

Por fim utilizando o código exemplo presente na listagem 3.1 é possível efetuar medições utilizando os quatro sensores presentes.

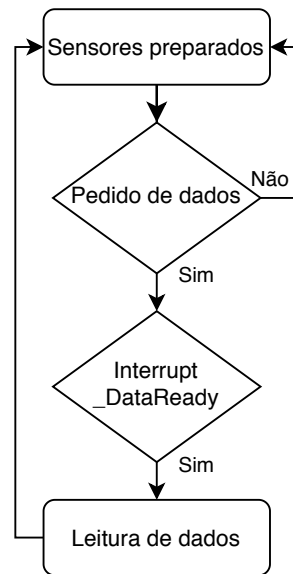


Figura 3.8: Fluxograma referente ao funcionamento do bloco utilizado para fazer a interface com os sensores DS18B20.

Utilizando a função *DS18B20_SendTemperatureRequest()* é possível enviar um pedido de conversão a todos os sensores conectados ao bloco. Após isto resta apenas esperar que os dados estejam preparados utilizando a *flag DS18B20_DataReady* e recolher os mesmos utilizando *DS18B20_ReadTemperature()* e *DS18B20_GetTemperatureAsFloat()*.

Listagem 3.1: Exemplo de medição de temperaturas.

```

1 #include "project.h"
2 float T1,T2,T3,T4 = 0; //Variaveis globais
3
4 int main(void){
5     CyGlobalIntEnable; //Interrupts globais
6     DS18B20_Start(); //Ativar bloco DS18B20
7     for(;;) //loop principal
8     {
9         DS18B20_SendTemperatureRequest();
10        while(!DS18B20_DataReady){}
11        DS18B20_ReadTemperature();
12        T1 = DS18B20_GetTemperatureAsFloat(0);
13        T2 = DS18B20_GetTemperatureAsFloat(1);
14        T3 = DS18B20_GetTemperatureAsFloat(2);
15        T4 = DS18B20_GetTemperatureAsFloat(3);
16    }
17 }
  
```

3.3.2 Módulo de monitorização de potência

Como uma das características mais importantes a monitorizar, senão a mais importante, temos a potência consumida pelo equipamento. Sendo assim, é necessário arranjar uma

CAPÍTULO 3. PROJETO DA ARQUITETURA E IMPLEMENTAÇÃO DO SISTEMA DE MONITORIZAÇÃO PARA PCM

solução simples, efetiva e de baixo custo com a capacidade de monitorizar no mínimo tensão, corrente e fator de potência, grandezas estas que podem ser posteriormente utilizadas para calcular a potência consumida pelo equipamento.

Adicionalmente funções como, medição automática de potência ativa, aparente, reativa e frequência, bem como, protocolos de comunicação I2C ou SPI são chave para um sistema de monitorização eficiente.

Para tal, foi escolhido o *Integrated Circuit* (IC) ACS71020 descrito em 2.5.3.3, visto que, este possui todas as características supracitadas. Além do mais são necessários pouquíssimos componentes externos, a gama de correntes suportadas é bastante ampla e os protocolos de comunicação são adequados a este tipo de sistemas, fazendo então com que a solução final seja de baixo custo e com a possibilidade de ter dimensões físicas reduzidas.

O ACS71020 é um IC de monitorização de potência de 16bits equipado com uma *Electrically-Erasable Programmable Read-Only Memory* (EEPROM) programável pelo utilizador, um condutor primário de $0.85m\Omega$ de modo a reduzir as perdas por efeito de joule e isolamento reforçado até $517V_{rms}$, certificado pela *Underwriters Laboratories* (UL).

Este IC, representado no diagrama de blocos presente na figura 3.9, utiliza tecnologia de medição de corrente em modo diferencial com isolamento galvânico baseada em efeito de Hall. Estas características fazem com que o IC não só, tenha a capacidade de rejeitar erros introduzidos por campos magnéticos em modo comum, mas também evite o uso de componentes caros e de dimensões elevadas como bobinas de rogowski, transformadores de corrente, OpAmp's isolados ou a perda de energia em resistências *shunt*.

Para efetuar uma medição de tensão este IC recorre ao uso de um divisor de tensão, de modo a reduzir a tensão de entrada nos pinos V_{in_p} e V_{in_n} a um máximo de $275mV$. Após reduzida a tensão à entrada do IC é feita a polarização da entrada V_{in_n} para que posteriormente se possa medir a diferença de potencial entre as duas entradas recorrendo ao uso de amplificadores de instrumentação.

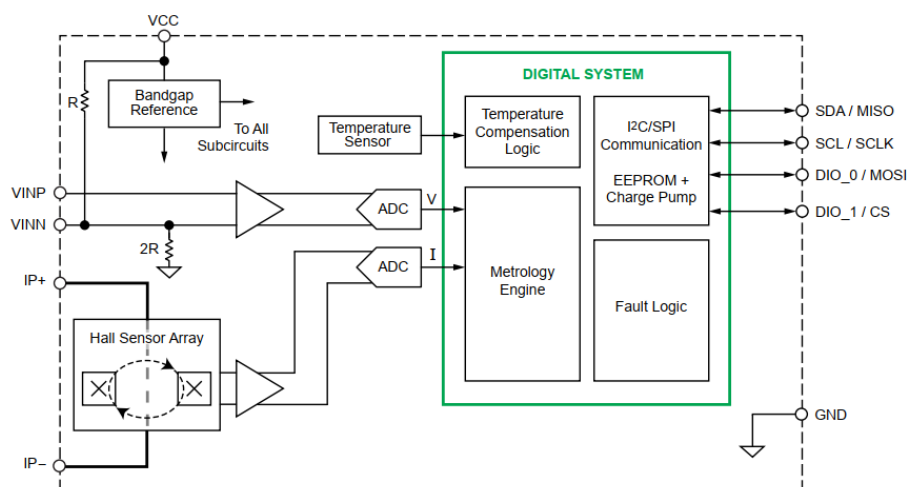


Figura 3.9: Diagrama de blocos funcional referente ao ic ACS71020.

Observa-se também na figura 3.9 que as duas medições analógicas são convertidas para o domínio analógico por via de dois ADC's e introduzidas no bloco *Metrology Engine*. Este bloco tem como função a correção e compensação, devido a fenómenos térmicos, de ambas as medidas de acordo com os dados fornecidos pelo bloco *Temperature Compensation Logic*. Após corrigidas as medições estas encontram-se prontas a serem utilizadas nos restantes cálculos e guardadas em registos que podem ser lidos pelo utilizador.

3.3.2.1 Implementação

De modo a efetuar medições de tensão eficaz é necessário recorrer a um divisor de tensão de modo a limitar a tensão de entrada para um valor máximo de $275mV$. Para tal foi utilizado o esquema aconselhado no *datasheet* presente na figura 3.10 . De acordo com [39] é aconselhado deixar uma margem de 40% da escala total, fazendo assim com que a queda de tensão aconselhada na resistência de medição seja de aproximadamente $165mV$.

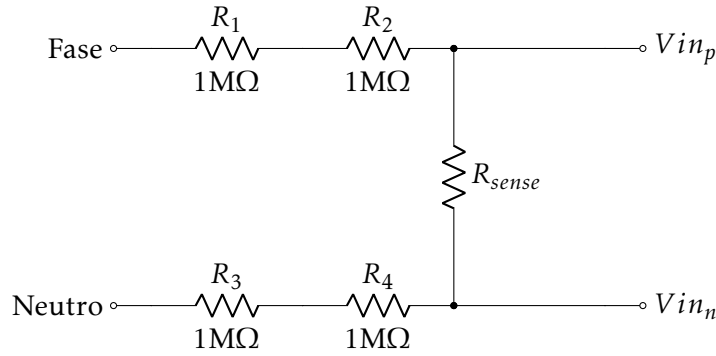


Figura 3.10: Divisor de tensão para leitura de tensão utilizando ACS71020.

Desenvolvendo a equação 3.1, referente ao divisor de tensão, é possível obter um valor aproximado da resistência R_{sense} a ser utilizada. Onde V_{sense} é a queda de tensão desejada na resistência R_{sense} e V_{in} é o valor de pico da tensão proveniente da rede que se pretende medir.

$$V_{sense} = \frac{R_{sense}}{R_{sense} + (R_1 + R_2 + R_3 + R_4)} \cdot v_{in} \quad (3.1)$$

Como R_1, R_2, R_3 e R_4 são conhecidas podem ser agrupadas, fazendo $R_T = R_1 + R_2 + R_3 + R_4$ obtém-se:

$$V_{sense} = \frac{R_{sense}}{R_{sense} + R_T} \cdot v_{in} \quad (3.2)$$

Desenvolvendo a expressão em ordem a R_{sense} obtém-se a seguinte expressão:

$$R_{sense} = \frac{V_{sense} \cdot R_T}{v_{in} + V_{sense}} \quad (3.3)$$

Sabendo que se deve deixar uma margem de 40% da escala total pode-se calcular então V_{sense} que será por sua vez os restantes 60%, V_{in} e R_T são também valores conhecidos

sendo estes valor de pico da tensão da rede de baixa tensão e a soma das restantes resistências presentes no circuito.

$$\begin{cases} V_{sense} = 0,6 \cdot 275 \cdot 10^{-3} = 165,6 \times 10^{-3} V \\ V_{in} = \sqrt{2} \cdot 230 V \\ R_T = 4 \times 10^6 \Omega \end{cases} \quad (3.4)$$

Posto isto é então possível substituir os valores conhecidos de modo a obter o valor da resistência R_{sense} :

$$R_{sense} = \frac{165,6 \times 10^{-3} \cdot 4 \times 10^6}{\sqrt{2} \cdot 230 + 165,6 \times 10^{-3}} = 2035,43 \Omega \quad (3.5)$$

Visto que o valor da resistência R_{sense} não deve ser superior a $2035,43 \Omega$, pois uma resistência de um valor superior se traduz numa maior queda de tensão em V_{sense} , escolheu-se como valor a utilizar $2000,0 \Omega$ sendo este um valor mais comum, que por sua vez se traduz numa maior facilidade de se adquirir.

Agora que a resistência de *sense* se encontra dimensionada é necessário calcular o valor de pico máximo de tensão que se pode medir e o pico máximo de potência ativa que este IC suporta. Estes valores são necessários, pois o IC não retorna valores concretos das medições que efetua, mas sim valores referentes a um fator de escala da grandeza que está a medir.

Utilizando assim a equação 3.2, resolvendo a mesma em ordem v_{in_max} e substituindo V_{sense} por os $275mV$, mencionado anteriormente é possível calcular o pico de tensão máxima que se pode medir com este IC.

$$v_{in_max} = V_{sense} \cdot \left(1 + \frac{R_T}{R_{sense}}\right) = 275 \times 10^{-3} \cdot \left(1 + \frac{4 \times 10^6}{2 \times 10^3}\right) = 550,275 V \quad (3.6)$$

Sabendo que o IC que foi adquirido tem um limite de $30A(I_{pr})$ de medição, e possuindo agora o valor de pico da tensão máxima que o mesmo é capaz de medir é então possível calcular o pico máximo de potência ativa, utilizando a formula da potência 3.6.

$$P_{max} = v_{in_max} \cdot I_{pr} = 550,275 \cdot 30 = 16508,25 W \quad (3.7)$$

Após definidos os parâmetros de funcionamento e calculados os dados necessários é necessário realizar a interface entre o sensor e o MCU. Para tal é necessário importar o bloco referente ao protocolo de comunicação SPI presente na figura 3.11, sendo que neste caso é necessário um *master* com um nível lógico de $3,3V$, pois o IC que foi adquirido não é tolerante a $5V$. De maneira a conseguir um nível lógico de $3,3V$ foi utilizado em DAC de tensão, configurado para essa mesma tensão, conectado aos pinos de saída escolhidos para a comunicação SPI.

Por último é necessário configurar o bloco de SPI para um modo compatível com o IC, para tal é necessário o modo onde a fase de *Clock* (CLK) e a polaridade de CLK são

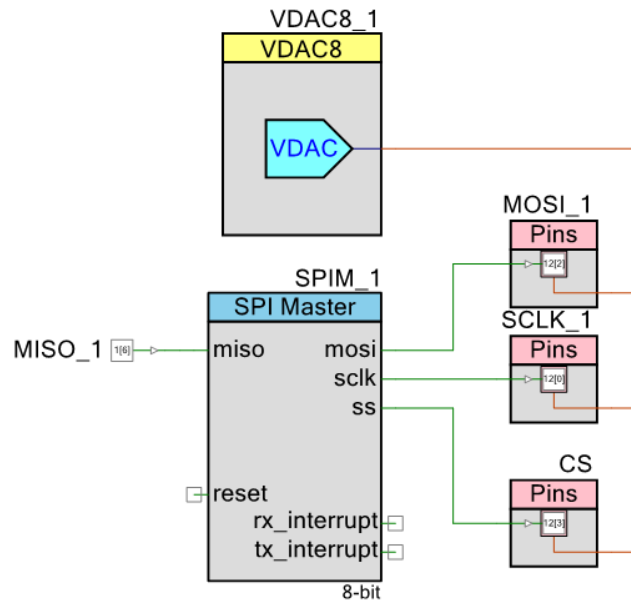


Figura 3.11: Bloco de comunicação SPI com tensão de referência de 3,3V gerada por DAC de tensão.

iguais a 1 com uma frequência compreendida entre 1MHz e 10MHz com 8bits de dados em formato *Least Significant Bit* (LSB).

Estando agora todas as configurações feitas resta então realizar uma leitura de dados do IC. De maneira a efetuar a leitura de um registo presente no IC basta enviar recorrendo as funções pertencentes ao bloco de SPI, o endereço do registo em questão alterando o bit mais significativo para 1 ou 0 caso se pretenda uma leitura ou escrita respetivamente, sendo necessário a função "*SPIM_1_WriteTxData(Address)*".

Feito o pedido de leitura o IC ira retornar uma resposta de 32bits, pela linha MISO, com o conteúdo do registo que se pretende ler, para se efetuar a leitura desta resposta é necessário criar uma rotina que faça quatro leituras consecutivas de 8bits, de modo a perfazer 32bits lidos, utilizando a seguinte função "*SPIM_1_ReadRxData()*". Na seguinte tabela 3.1 é possível observar os registos referentes as leituras de tensão e corrente RMS, bem como as potências, o fator de potência, o tipo de valor que se pode obter e o fator de escala.

Tabela 3.1: Registos do ic ACS71020, grandezas correspondentes e fator de escala

Grandeza	Endereço	Tipo	Intervalo	Factor de escala
Tensão	0x20	15bits fracionários sem sinal	[0; ~1[v_{in_max}
Corrente	0x20	15bits sem sinal com 14 bits fracionários	[0; ~2[I_{p_r}
Potência Activa	0x21	17bits com sinal com 15 bits fracionários	[-2-, 2[P_{max}
Potência Aparente	0x22	16bits sem sinal com 15 bits fracionários	[0; 2[P_{max}
Potência Reactiva	0x23	16bits sem sinal com 15 bits fracionários	[0; 2[P_{max}
Factor de Potência	0x24	11bits com sinal com 9 bits fracionários	[-2; 2[N/A

3.4 Módulo de aquisição de dados

O módulo de aquisição de dados tem como função fazer a interface com os sensores que a ele estão ligados e recolher dados provenientes dos mesmos, para posteriormente serem processados e recolhidos pelo módulo de comunicação.

Visto que este modulo estará apenas encarregue de recolher e processar dados é necessário fazer a escolha de um MCU com algum poder de processamento e boas ferramentas de *hardware* e *software* ao seu dispor. Porém características como *Wi-Fi* e *Bluetooth*, bem como capacidade de captura e processamento de vídeo, vêm-se como desnecessárias devido à função primária deste módulo e ao facto de esta solução ainda se encontrar numa fase de prototipagem.

Na subsecção seguinte ir-se-á mais ao detalhe sobre da escolha do MCU que melhor se adequa, não só pelas suas funções, mas também pelo seu fator custo/benefício e sobre o funcionamento do módulo como um todo.

3.4.1 Escolha do MCU

Atendendo aos MCU's presentes na subsecção 2.5.1.5 determinou-se que o Raspberry Pi 3B+ não é adequado para pertencer a este bloco visto que, o seu consumo de potência é elevado comparativamente aos outros MCU's, o seu custo base é o mais elevado de todos os MCU's presentes e não possui ADC's, fazendo com que a sua implementação seja mais difícil resultando num custo final mais elevado.

Apesar de o MCU ESP32 possuir todas as capacidades necessárias para fazer as funções não só do módulo de aquisição de dados, bem como simultaneamente, fazer as funções do módulo de comunicação, decidiu-se que o ESP32 seria apenas utilizado no módulo de comunicações resultando assim numa arquitetura mais versátil como foi explicitado anteriormente.

Restando dois MCU's na tabela 2.8 é necessário fazer uma comparação mais detalhada, de modo a escolher o MCU que melhor se adequa tanto as necessidades presentes do módulo de aquisição de dados bem como as necessidades futuras e os possíveis sensores com que poderá vir a fazer interface com.

Primeiramente tem-se o Arduino Uno, esta placa de desenvolvimento *open-source* está equipada com um MCU de *8bits*, com uma frequência de funcionamento até *16MHz* possuindo também, *32KB* de memória programável um módulo de comunicação *Universal Asynchronous Receiver-Transmitter* (UART), dois módulos de comunicação SPI e I2C e por fim um ADC de *10bits* e seis canais.

Em segundo tem-se o PSoC 5Lp, esta placa de desenvolvimento faz uso de um processador de *32bits* Arm Cortex-M3 com uma frequência até *80MHz*, tem a capacidade de ter até 6 níveis de tensão interna, *256KB* de memória programável, um ADC Sigma-Delta de *20bits* e um ADC SAR de *12bits*, uma corrente de *2μA* em modo *sleep* e de *300nA* em modo de hibernação e por último a capacidade de se importar múltiplos blocos SPI e I2C

em simultâneo e em diferentes modos, quer sejam, *master* ou *slave*.

Esta ultima característica é bastante importante pois pretende-se, numa fase mais avançada do projeto que existam vários módulos de monitorização ligados a apenas um módulo de comunicação, fazendo assim com que seja necessário efetuar uma comunicação I2C *slave* com o módulo de comunicação e uma possível comunicação I2C *master* os módulos de sensores. Atendendo as estas características e à fase do projeto, o Arduino Uno encontra-se como uma opção bastante viável no entanto, o facto de apenas ter um módulo de comunicação I2C e o seu preço ser superior, por um fator de 2, ao do PSoC 5Lp faz com que a melhor placa de desenvolvimento a ser utilizada, do ponto de vista técnico e monetário, seja o PSoC 5Lp.

3.4.2 Implementação

Nesta subsecção pretende-se explicitar apenas o funcionamento do modulo de aquisição de dados e as interações pretendidas com o módulo de comunicação, visto que, o funcionamento do modulo de aquisição de dados com os módulos de sensores já se encontra abrangido nas secções anteriores.

Atendendo às características da arquitetura que se pretendem implementar será necessário um protocolo de comunicação endereçável, bidirecional e simples de implementar. Para tal, foi decidido que a comunicação que mais se adequava seria a comunicação I2C, pois esta tem uma grande vantagem relativamente à alternativa a comunicação SPI, sendo esta vantagem o método de endereçamento dos dispositivos.

No caso da comunicação SPI o endereçamento é feito utilizando o sinal de *Chip-Select* (CS) onde é apenas ativado *1bit* endereçando apenas um dispositivo, isto significa que serão necessários tantos sinais de CS quanto o numero de dispositivos no barramento SPI.

Em contrapartida, na comunicação I2C o endereçamento é feito a partir do barramento de dados, *Serial Data* (SDA), onde o primeiro *byte* a ser enviado no barramento contem o endereço de *7bits* do dispositivo alvo mais *1bit* sinalizando se a comunicação que se pretende efetuar é uma leitura ou uma escrita. Por último, ao possuir um endereço de *7bits* em teoria é possível ter até 127 dispositivos em apenas uma linha I2C. Posto isto, resta então importar o bloco presente no PSoC responsável pela comunicação I2C como se pode observar na figura 3.12.

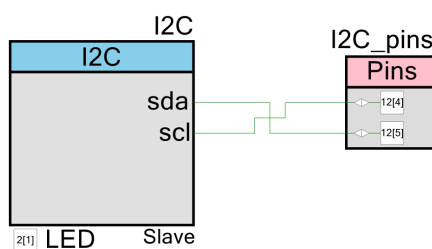


Figura 3.12: Bloco I2C slave presente no PSoC 5Lp.

Após configurar o endereço ao qual este módulo irá responder será necessário inicializar dois *buffers* um para o envio e outro para a recepção de dados e por fim, conectar as linhas SDA e *Serial Clock* (SCL) a dois pinos bidirecionais, configurados em modo de coletor aberto de maneira a fazer *drive* de ambas as linhas a *ground* sinalizando assim *1bit*.

Configurada a comunicação I2C é necessário programar o comportamento deste módulo de aquisição de dados para que consoante o comando recebido, proveniente do modulo de comunicação, efetue e armazene medições dos diferentes sensores para posteriormente serem solicitados pelo modulo de comunicação. No fluxograma presente na figura 3.13 é possível observar o funcionamento base do modulo de aquisição de dados.

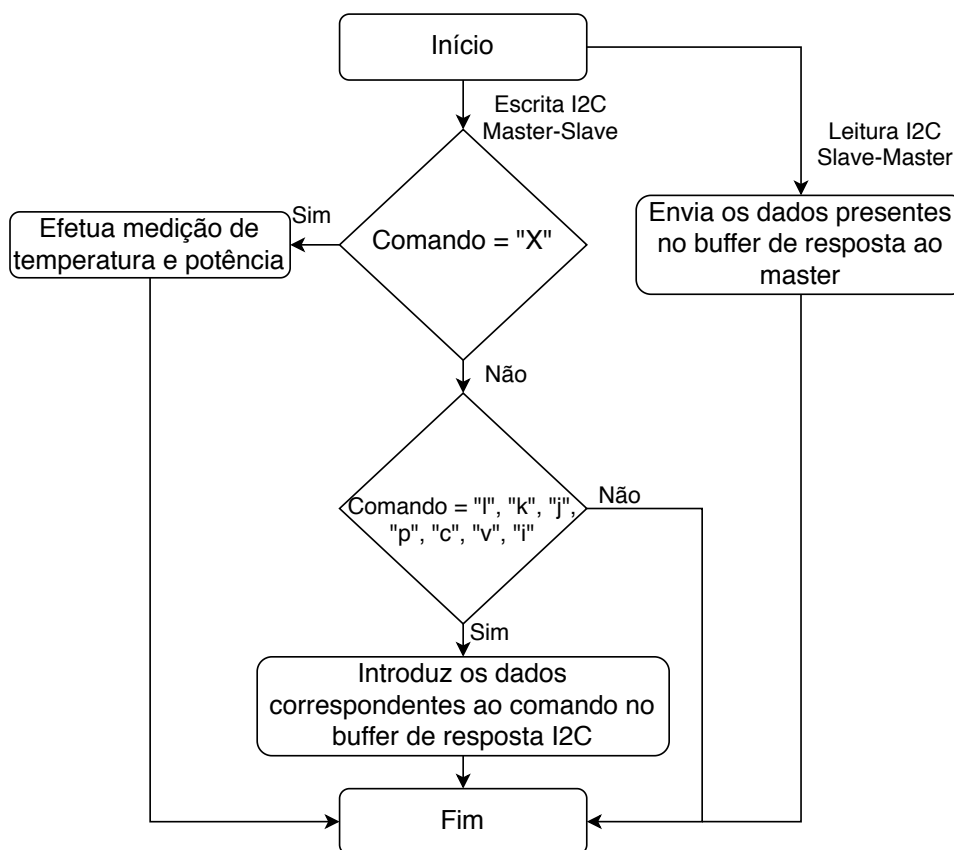


Figura 3.13: Fluxograma referente ao comportamento do módulo de aquisição de dados.

Este fluxograma primeiramente encontra-se em um estado de espera até que se efetue uma escrita ou uma leitura, no caso de se receber o comando "x" ir-se-á efetuar uma medição de todos os sensores presentes, caso se receba outro comando irá introduzir os dados no *buffer* de envio de dados, ou simplesmente ignorar se o comando não for reconhecido. Ao receber um pedido de leitura serão enviados os dados previamente introduzidos no *buffer*.

3.5 Módulo de comunicação

O módulo de comunicação tem como função coletar os dados recolhidos pelos módulos de monitorização presentes no sistema, para posteriormente serem enviados para uma plataforma IoT, de modo a serem armazenados e exibidos de uma forma simplificada. Sendo assim, de grosso modo, o módulo de comunicação é uma "ponte" entre os módulos de monitorização e a plataforma IoT devido ao facto de apenas coletar informação e enviar essa mesma informação através de um protocolo de mensagens web para a plataforma escolhida.

Para realizar esta função foi escolhido o MCU ESP32, visto que este tem uma maior facilidade de integração e programação, graças as funcionalidades de *Wi-Fi* e *Bluetooth* incluídas e a capacidade de utilizar como ambiente de programação o IDE *Arduíno*, sendo possível a compilação e programação do MCU através da porta USB. Adicionalmente, o custo deste MCU é substancialmente menor que o custo do *Arduíno* e a sua performance superior.

3.5.1 Implementação

Atendendo à especificação de que a comunicação entre o módulo de aquisição de dados e o módulo de comunicação deve ser I2C, recorreu-se à biblioteca "*Wire.h*" que permite a declaração de pinos I/O a serem utilizados na comunicação, nomeadamente SDA e SCL e a alteração da frequência de *clock* presente na linha de SCL. Posto isto, resta a criação de algumas rotinas com as funções de escrita e leitura I2C, de maneira a que a implementação de rotinas de maior dimensão e complexidade seja simplificada.

Na listagem 3.2 é possível observar duas rotinas, sendo que a primeira tem como função a escrita de informação para um dispositivo I2C *slave* e a segunda tem como função a leitura de dados. Estas rotinas utilizam funções como *Wire.beginTransmission()*, *Wire.write()*, *Wire.requestFrom()*, *Wire.available()* e *Wire.endTransmission()* para começar uma transmissão, escrever, fazer um pedido de leitura, verificar se existe atividade por parte de um slave e terminar a transmissão respetivamente.

Listagem 3.2: Rotinas de escrita e leitura I2c.

```
1 //escrita de um carácter master-slave
2 static void Data_Write(char f, int addr){
3     Wire.beginTransmission(addr);
4     Wire.write(f);
5     Wire.endTransmission();}
6 //leitura de dados slave-master
7 static void Data_Read(int addr){
8     Wire.requestFrom(addr,5);
9     while( Wire.available() ){
10         x = Wire.read();
11         response += x;}}
```

A interação entre do módulo de comunicação e a plataforma IoT será feita através de um protocolo de mensagens web, neste caso *MQ Telemetry Transport* (MQTT) operando sobre *Wi-Fi*, de maneira a tomar partido das funcionalidades presentes no MCU ESP32.

Visto que a plataforma a utilizar será a plataforma *Azure for IoT*, recorreu-se a uma biblioteca específica para o ESP32 onde os protocolos de comunicação necessários já se encontram configurados, de modo a estabelecer uma conexão com a plataforma de uma forma rápida e segura.

Para estabelecer a ligação entre o módulo e a plataforma é apenas necessário utilizar uma rotina de conexão pertencente à biblioteca, introduzindo na mesma a chave única de conexão à plataforma e o *ID* único associado a essa chave, que se encontram disponíveis na listagem de dispositivos autorizados na plataforma. Após estabelecida a conexão basta organizar os dados provenientes do módulo de monitorização num formato aceite pela plataforma, seja este *JavaScript Object Notation* (JSON) ou *Comma-separated values* (CSV) e fazer o envio dos mesmos através de uma outra rotina pertencente à biblioteca.

Estando definidas e configuradas as funcionalidades básicas do módulo de comunicação é necessário definir o comportamento do módulo como um todo para que este realize a sua função. Para tal, recorre-se ao uso de um fluxograma apresenta na figura 3.14 para definir o seu comportamento.

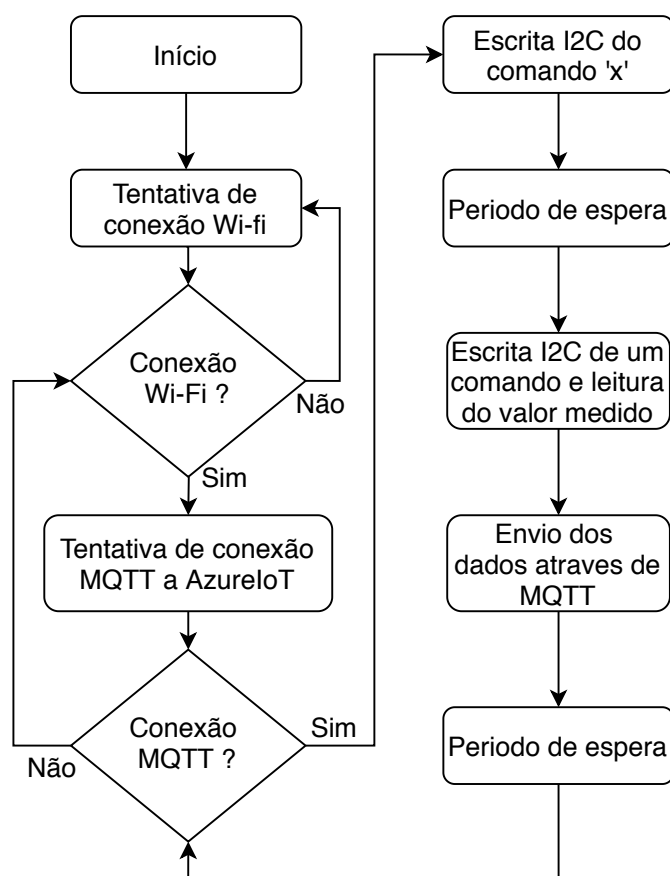


Figura 3.14: Fluxograma do funcionamento do módulo de comunicação.

No fluxograma tem-se o comportamento desejado para o módulo de comunicação. Partindo do início o módulo tem de assegurar que possui uma conexão sólida à Internet através de *Wi-Fi*, após conectado à Internet é necessário estabelecer uma ligação através de MQTT à plataforma para que posteriormente seja possível enviar os dados recolhidos. Caso alguma destas duas tentativas de conexão falhe não será possível que o módulo efetue a sua função, sendo assim é necessário continuar a realizar tentativas até que sejam estabelecidas as ligações com sucesso.

Estando conectado à Internet e à plataforma é então necessário enviar um comando de sinalização para que o módulo de aquisição de dados efetue medições necessárias, sendo que o comando que sinaliza esta ação é o carácter "x". Seguidamente, será necessário efetuar um período de espera de aproximadamente 800ms.

A razão para a qual se efetua um período de espera neste momento é devido ao facto de alguns sensores, como os sensores de temperatura, necessitarem de tempo para efetuar a conversão analógica digital e o MCU presente no módulo de aquisição de dados necessitar de efetuar medições sobre diversos sensores, com a possibilidade de realizar algum processamento sobre os dados. Ao definir um período de amostragem é imperativo ter em conta este primeiro período que não deve ser inferior a 800ms. Posto isto, o período de amostragem é então a soma do primeiro período de espera mais o segundo período presente no fluxograma.

Realizado o primeiro período de espera efetua-se a escrita de um comando referente ao valor que se pretende obter e seguidamente faz-se a leitura desse mesmo valor. Nesta fase é possível efetuar a leitura de vários valores, para que sejam enviados em simultâneo na mesma mensagem JSON. Após enviados os dados realiza-se mais um período de espera, para que a próxima mensagem a ser enviada cumpra o período de amostragem pretendido e retorna-se ao ponto do fluxograma onde se verifica se existe conexão MQTT criando assim um ciclo de verificação de conexão e envio de dados para a plataforma.

3.6 Plataforma IoT

Pretende-se ao implementar uma plataforma IoT que esta suporte algumas características chave sendo estas, a capacidade de estabelecer uma ligação segura entre o dispositivo IoT e a plataforma, a possibilidade de gerir diferentes dispositivos IoT de modo a ter um influxo de dados estável e organizado, a capacidade de fazer roteamento de dados seletivo, a possibilidade de integração com diversas outras plataformas e funcionalidades de modo a armazenar os dados recebidos e exibir os mesmo num formato lógico e de fácil compreensão para o utilizador final e por fim o custo associado a estes serviços.

Atendendo as características necessárias existem três serviços que se destacam, sendo estes a *AWS IoT core*, *Azure for IoT* e a *Google Cloud IoT Core* pois possuem mais que um tipo de ferramenta para gestão de dispositivos, protocolos de comunicação sendo o MQTT suportado em todas elas, diversos tipos de armazenamento seja este frio (com pouco acesso aos dados e uma longa duração de armazenamento) ou quente (acesso muito

frequente a dados com duração de armazenamento indefinida) e integração com várias outras plataformas [40].

Visto que todas estas plataformas possuem as capacidades necessárias resta então analisar o custo associado a cada uma. De acordo com [41] é possível constatar que a *Azure for IoT* é a plataforma que mais serviços grátis possui, será também importante mencionar que a *AWS IoT core* é a única plataforma que não possui um serviço grátis para a conexão de apenas um dispositivo IoT. Posto isto a *Azure for IoT* é a plataforma que mais simplifica a integração de serviços IoT para um projeto na fase de prototipagem.

Adicionalmente, existe um programa para estudantes que pretendam utilizar a plataforma *Azure for IoT*. Este programa visa que os estudantes tenham a capacidade de experimentação, mesmo que limitada, de funcionalidades que não se encontrem grátis ao fornecer um crédito de 100\$ que poderá ser gasto em qualquer funcionalidade da plataforma ao longo de 12 meses. Por estas razões a plataforma *Azure for IoT* foi escolhida como a plataforma a utilizar, pois possui o maior numero de funcionalidades e integrações mantendo simultaneamente um custo bastante competitivo.

3.6.1 Implementação

De modo a satisfazer as necessidades supracitadas é necessário utilizar vários tipos de serviços, na figura 3.15 é possível observar todos os serviços utilizados e as ligações entre os mesmos.

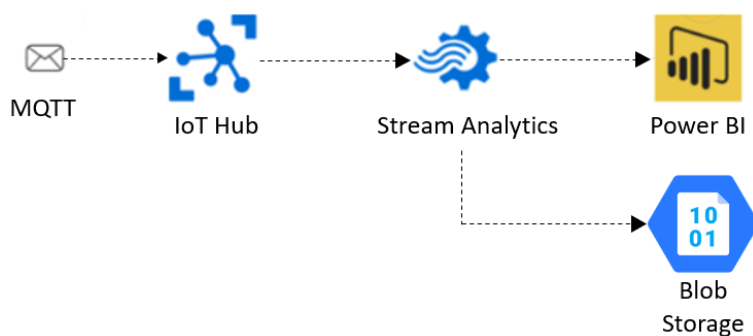


Figura 3.15: Serviços utilizados na plataforma IoT.

Primeiramente, para conectar qualquer tipo de dispositivo IoT será necessário a implementação do serviço *IoT Hub*, figura 3.16, este serviço habilita uma comunicação segura e fiável com os dispositivos, possuindo apenas um canal de *output* contendo todos os dados recebidos. Adicionalmente este serviço possui algumas funções de gestão e aprovisionamento, de modo ligar e gerir os dispositivos IoT conectados.

Ao criar este serviço é necessário ter em atenção o nome que será dado ao *Resource Group*, apesar de a utilização de caracteres como espaços ser autorizada estes podem criar conflitos ao serem utilizados com os serviços seguintes, a região onde este se encontra é também importante estar correta visto que possui um grande impacto na latência do serviço e nos custos associados à transferência de dados entre servidores.

IoT hub

Microsoft

Basics Networking Size and scale Tags Review + create

Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. [Learn more](#)

Project details

Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription * ⓘ Azure para Estudantes

Resource group * ⓘ PcmDataGroup
[Create new](#)

Region * ⓘ West Europe

IoT hub name * ⓘ pcmdatastorage ✓

Figura 3.16: Criação do serviço IoT Hub.

Criado o serviço é então necessário adicionar um dispositivo, para que seja gerada uma chave única de acesso primário, bem como um *ID* único associado ao dispositivo. Após aceder ao menu *IoT devices* e adicionar um dispositivo, obtêm-se algumas chaves de acesso, na figura 3.17 encontra-se delineada a vermelho a chave que é necessário adicionar ao módulo de comunicação descrito anteriormente. Estando adicionado um dispositivo IoT será necessário aceder ao menu *Built-in endpoints*, para que seja criado um *Consumer Group* que terá como função a associação dos dados recebidos num grupo definido.

esp32 ⓘ

Save Message to Device Direct Method Add Module Identity Device Twin Manage keys Refresh

Device ID	esp32
Primary Key	NBmdusW5qgETacCgDmdT/8iggaqdrpW5oFolLagx
Secondary Key	sUX94ub1534Wbaf8/7q/tpuLag1dmKyhbfQ39L8x
Primary Connection String	HostName=PcmHubIoT.azure-devices.net;DeviceId=esp32;SharedAccessKey=NBmdusW5qgETacCgDmdT/8iggaqdrpW5oFolLagx
Secondary Connection String	HostName=PcmHubIoT.azure-devices.net;DeviceId=esp32;SharedAccessKey=sUX94ub1534Wbaf8/7q/tpuLag1dmKyhbfQ39L8x

Enable connection to IoT Hub ☒ Enable ☐ Disable

Parent device No parent device

Module Identities Configurations

MODULE ID	CONNECTION STATE	CONNECTION STATE LAST UPDATED (UTC)	LAST ACTIVITY TIME (UTC)
There are no module identities for this device.			

Figura 3.17: Chave de acesso primária.

Estando criado um serviço que permite a conexão de dispositivos IoT é necessário adicionar os restantes serviços, no caso de se pretender apenas um tipo de funcionalidade desta plataforma poder-se-ia conectar diretamente o serviço associado a essa funcionalidade ao *IoT Hub*, no entanto neste caso pretende-se que a plataforma tenha a capacidade de exibir e armazenar os dados recebidos simultaneamente.

Sendo que são necessários dois tipos de serviços, será necessário adicionar um mecanismo com a capacidade de criar vários canais de *output* de dados ou neste caso *pipeline's*. Para tal, é necessário utilizar o serviço *Stream Analytics*, sendo que este serviço permite a criação de *pipeline's* ponto-a-ponto onde é possível efetuar *query's* sobre os dados a serem

CAPÍTULO 3. PROJETO DA ARQUITETURA E IMPLEMENTAÇÃO DO SISTEMA DE MONITORIZAÇÃO PARA PCM

enviados utilizando *Structured Query Language* (SQL), tendo a possibilidade de adicionar funcionalidades personalizadas utilizando *JavaScript* ou *C#*. Ao criar o serviço *Stream Analytics* é necessário ter em atenção, à semelhança do *IoT Hub* o *Resource Group* que neste caso já se encontra criado e a localização do servidor.

Após introduzido o mecanismo que permite a transmissão de dados para diferentes destinos, será então necessário escolher e ativar os serviços que vão desempenhar as funções pretendidas nesta plataforma IoT. Para a visualização de dados foi utilizada a ferramenta *PowerBI*, contendo uma coleção de serviços de software e aplicações que funcionam em conjunto para converter dados não relacionados em informação coerente e causal, visualmente envolvente e interativa. Quanto ao armazenamento de dados será utilizado o serviço *Blob Storage*, visto que este se encontra otimizado para armazenar quantidades em grande escala de dados não estruturados. De modo a utilizar o serviço de *Blob Storage* é necessário a criação de uma conta de armazenamento, na figura 3.18, encontram-se as definições utilizadas para a criação da mesma.

Create storage account

Basics Networking Data protection Advanced Tags Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below. [Learn more about Azure storage accounts](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure para Estudantes

Resource group * PcmDataGroup [Create new](#)

Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

Storage account name * blobstoragejob ✓

Location * (Europe) West Europe

Performance ☒ Standard ☐ Premium

Account kind ☐ StorageV2 (general purpose v2)

Replication ☐ Read-access geo-redundant storage (RA-GRS)

Figura 3.18: Criação de uma conta de armazenamento.

Os dados gerados pelo nó IoT são considerados não estruturados devido ao facto de alterarem consideravelmente consoante o sistema que se pretende monitorizar, e.g. para um sistema de dimensões reduzidas poderá ser necessário apenas um sensor de temperatura enquanto que, num sistema de dimensões elevadas poderão ser necessários vários sensores de temperatura, alterando assim a estrutura dos dados que são enviados para a plataforma IoT.

Estando criada a conta de armazenamento é então necessário indicar ao serviço o formato de dados que se pretende armazenar, sendo o formato não estruturado é necessário

a adição de um *container*.

Estando todos os serviços criados é necessário adicionar ao *Stream Analytics* o *IoT Hub* como um serviço de entrada, visto que este é a fonte de dados, e os serviços *Blob Storage* e *PowerBI* serão adicionados da mesma forma no entanto serão serviços de saída.

Para a adição destes serviços é necessário aceder ao menu *Job topology* selecionando *inputs* ou *outputs* consoante o que se deseja adicionar. Em relação ao *Power Bi* basta efetuar a vinculação do serviço, sendo que não serão necessárias configurações adicionais. Quanto aos dois restantes serviços que se pretendem adicionar, será necessário definir algumas configurações, para o *IoT Hub* é necessário adicionar o *Consumer group* criado anteriormente e definir o *Shared access policy name* de modo a que o nome do serviço esteja seguido de "owner". Para a adição de um serviço *Blob Storage* é necessário a utilização do *container* criado previamente, na figura 3.19, encontram-se as definições utilizadas em detalhe.

The image shows two side-by-side configuration panels for Stream Analytics.

IoT Hub (New input):

- Input alias: `IoTHubLinup` (with a green checkmark)
- Options: ☐ Provide IoT Hub settings manually; ☒ Select IoT Hub from your subscriptions
- Subscription: `Azure para Estudantes`
- IoT Hub: `PcmHubIoT`
- Endpoint: `Messaging`
- Shared access policy name: `iothubowner`
- Shared access policy key: (masked with asterisks)
- Consumer group: `streamanalytics_consumergroup`
- Partition key: (empty field)
- Event serialization format: `JSON`
- Encoding: `UTF-8`
- Event compression type: `None`

Blob storage/ADLS Gen2 (New output):

- Storage account: `pcmdatastorage`
- Storage account key: (masked with asterisks)
- Container: `storageoutput`
- Options: ☐ Create new; ☒ Use existing
- Path pattern: (empty field)
- Date format: `YYYY/MM/DD`
- Time format: `HH`
- Event serialization format: `JSON`
- Encoding: `UTF-8`
- Format: `Line separated`
- Minimum rows: (empty field)
- Maximum time: `Hours` (empty field) and `Minutes` (empty field)

Figura 3.19: Definições detalhadas sobre a adição dos serviços *IoT Hub* e *Blob Storage* ao *Stream Analytics*.

Por fim, de modo a efetuar o roteamento dos dados para os diferentes serviços é necessário a criação de uma *Query*, utilizando os serviços adicionados previamente aos

CAPÍTULO 3. PROJETO DA ARQUITETURA E IMPLEMENTAÇÃO DO SISTEMA DE MONITORIZAÇÃO PARA PCM

inputs e *outputs*. No anexo III é possível analisar a *Query* implementada para visualizar e armazenar os dados provenientes do módulo de comunicação.

RESULTADOS EXPERIMENTAIS

Neste capítulo serão apresentados alguns cenários de teste, realizados com o intuito de testar o desempenho e o funcionamento da arquitetura proposta no capítulo anterior. Após a validação da plataforma, será apresentado um ensaio onde se pretende estudar os efeitos da utilização de materiais PCM para o armazenamento térmico de energia.

4.1 Validação da plataforma

Os cenários visam o teste de alguns atributos específicos e de elevada importância para a caracterização das capacidades e limitações do sistema implementado, sendo os atributos: a **precisão** esperada pelo módulo de monitorização potência; a **estabilidade** do sistema ao efetuar testes de longa duração; a **sensibilidade** do sistema a estímulos exteriores; e os **pontos de funcionamento crítico** do sistema em diferentes ambientes.

Será também importante mencionar que para alguns destes testes seria ideal a utilização de equipamento de precisão elevada, no entanto, é possível obter dados bastante robustos combinando análises teóricas com cenários teste onde é possível diminuir os estímulos exteriores para um mínimo e a utilização de um equipamento de medição com uma precisão de medição de classe 1 bastante utilizada na indústria.

4.1.1 Cenário 1 - Teste controlo utilizando uma carga conhecida.

Neste primeiro cenário pretende-se efetuar um teste de controlo utilizando uma carga conhecida de modo a analisar a precisão que se pode obter com o IC ACS71020, utilizando a montagem descrita na figura 3.10. Adicionalmente, pretende-se testar o funcionamento dos sensores de temperatura DS18B20 e da plataforma *Azure for IoT*, traçando os dados armazenados pela plataforma de modo a observar a causalidade entre o consumo de energia e o aumento da temperatura da carga em função do tempo.

4.1.1.1 Sistema utilizado para a monitorização

Para tal, efetuou-se a montagem do módulo de monitorização de potência presente na figura 4.1. Esta montagem possui um regulador de tensão, de modo a converter os 5V provenientes do PSoC 5Lp para os 3,3V necessários para alimentar o ACS71020, possui também dois condutores de $0,75\text{mm}^2$, em paralelo, para a medição de corrente perfazendo assim uma secção de $1,5\text{mm}^2$ capaz de suportar uma intensidade de corrente até 19A e o divisor resistivo para efetuar medições de tensão.

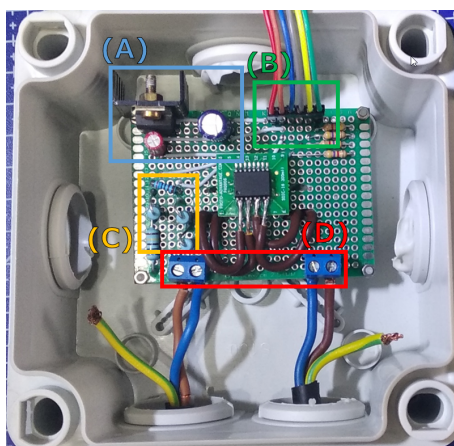


Figura 4.1: Módulo de monitorização de potência. (A) - Alimentação 3,3v, (B) - Comunicação e alimentação, (C) - Divisor resistivo, (D) - Bloco de terminais para Fase e Neutro.

Depois de efetuada a montagem do módulo de monitorização de potência conectaram-se os sensores de temperatura e o módulo ao PSoC 5lp. Posteriormente efetuou-se a ligação entre o módulo de monitorização e o módulo de comunicação tendo sempre em atenção os esquemas e métodos de conexão mencionados no capítulo anterior, obtendo assim o sistema presente na figura 4.2.

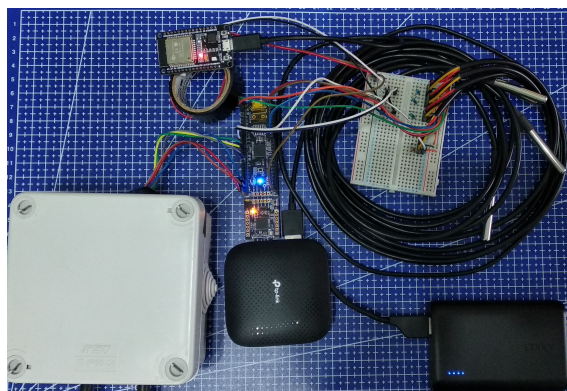


Figura 4.2: Montagem completa. Alimentação a partir de um *Power Bank*.

De modo a verificar se os valores de potência recolhida pelo módulo descrito anteriormente estão corretos utilizou-se o medidor de potência PM9C da *Schneider Electric*

presente na figura 4.3. Este medidor foi equipado com uma ficha monofásica para se conectar à rede de baixa tensão e duas tomadas monofásicas para que seja possível conectar os dispositivos a serem medidos.



Figura 4.3: Medidor de potência PM9C da *Schneider Electric*.

Para efetuar o teste de controlo foi escolhida como carga conhecida uma lâmpada de incandescência de 70W presente na figura 4.4, esta possui apenas um elemento resistivo no seu interior fazendo assim com que não existam consumos de corrente não lineares, visto que o seu fator de potência é aproximadamente 1, ou seja, uma carga puramente resistiva. A lâmpada foi conectada ao módulo de monitorização de potência utilizando o casquilho apropriado e para se obter uma leitura de temperatura foi colocado um sensor de temperatura o mais perto possível da lâmpada, na seguinte figura é possível observar a montagem descrita.



Figura 4.4: Lâmpada de incandescência de 70W com sensor de temperatura.

4.1.1.2 Análise dos dados obtidos

Agora que o cenário de testes se encontra estruturado, resta então proceder ao teste. Para tal, conectou-se a montagem à rede de baixa tensão, fazendo com que o medidor PM9C seja o primeiro equipamento de medição seguido pelo módulo de monitorização de potência e por fim a carga. Na figura 4.5 encontram-se representados os dados de potência e temperatura da carga em função do tempo.

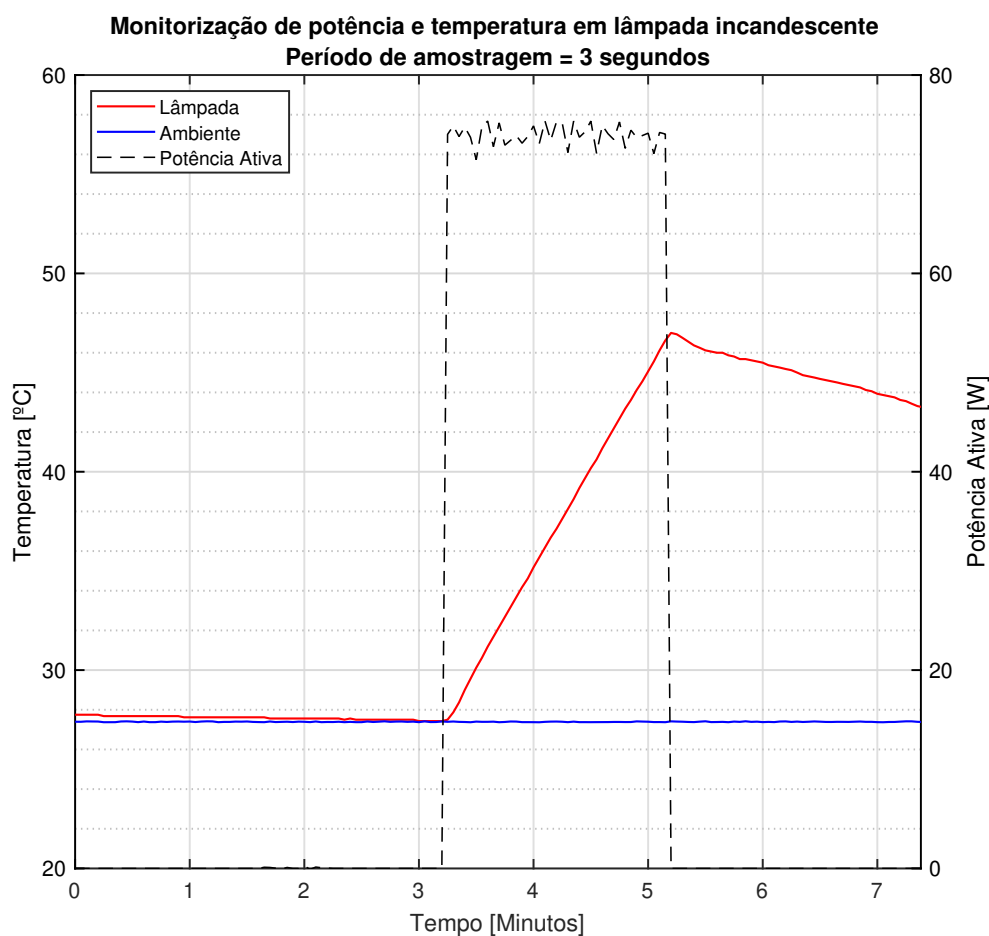


Figura 4.5: Consumo energético e temperatura de uma lâmpada de incandescência de 70W em função do tempo.

Ao analisar os dados recolhidos, é possível observar que a temperatura da lâmpada começa a aumentar no momento em que a mesma começa a consumir energia elétrica e diminui de uma forma mais lenta no momento em que a lâmpada é desligada, sendo assim possível retirar a simples conclusão que existe um efeito de causalidade entre o consumo energético numa carga e a sua temperatura. Este efeito por si só já seria esperado, no entanto com este teste foi possível observar que o módulo de aquisição de dados está a funcionar corretamente pois não existe nenhum comportamento fora do normal e o dados

aparentam estar sincronizados. Adicionalmente detetou-se que o consumo de potência da lâmpada de 70W se encontra relativamente elevado, aproximando-se dos 75W, este erro de medição poderá derivar de vários fatores como, o valor da tensão de alimentação não ser exatamente 230V, a existência de um erro intrínseco aos valores das resistências utilizadas no divisor resistivo responsável pela medição de tensão, um erro associado à medição de corrente por efeito de hall e o facto de a montagem efetuada em 4.1 ser relativamente sensível a ruídos e interferências exteriores.

Ao fazer o teste foi possível constatar, utilizando o medidor PM9C, que o valor da tensão (V_{real}) se encontrava ligeiramente elevado sendo este 233V. Recorrendo a um calculo simples é possível verificar que esta diferença na tensão de alimentação se converte num acréscimo de aproximadamente 1,83W ao valor nominal.

$$P = \frac{V^2}{R} \quad (4.1)$$

Utilizando a formula 4.1 é possível retirar o valor da resistência da lâmpada de incandescência, visto que o valor da resistência altera pouco com a temperatura é seguro ignorar esta variação.

$$R = \frac{V^2}{P} = \frac{230^2}{70} = 755,7143\Omega \quad (4.2)$$

Tendo agora o valor da resistência da lâmpada será possível calcular o valor real da potência consumida (P_{real}).

$$P_{real} = \frac{233^2}{755,7143} = 71,83W \quad (4.3)$$

Sabendo agora que a potência real consumida pela carga é de 71,83W, resta então calcular o erro associado à medição de potência de maneira confirmar se o pico de 75,3W estará dentro do intervalo de erro aceite. Para tal, foi realizada uma análise de monte carlo sobre o circuito presente na figura 3.10, onde se utilizou resistências com 1% de erro, obtendo-se como valor de pior caso possível uma tensão V_{sense} de 117,1mV quando são aplicados 230V no divisor resistivo, sendo que o valor teórico calculado é de 114,9mV. Posto isto é possível calcular que o erro relativo de medição da tensão é de aproximadamente 1,915%.

$$E_{r_V} = (117,1 \times 10^{-3} - 114,9 \times 10^{-3}) \cdot \frac{100}{114,9 \times 10^{-3}} = 1,915\% \quad (4.4)$$

Atendendo ao *datasheet* do IC ACS71020 é possível verificar que o erro relativo da medição de corrente (E_{r_I}) é de aproximadamente 3%. Tendo agora dados suficientes relativamente aos erros associados às medidas efetuadas é então possível calcular o erro associado à medição de potência (E_{r_Pot}). Sendo a potência o produto de duas grandezas o seu erro relativo será a soma dos erros relativos respetivos ao produto (E_{r_V} e E_{r_I}).

$$E_{r_Pot} = E_{r_V} + E_{r_I} = 1,915 + 3 = 4,915\% \quad (4.5)$$

De modo a verificar se o pico de $75,3W$ estará dentro do intervalo de erro aceite, será necessário utilizar a potência real consumida e o erro relativo associado à medição de potência para obter um intervalo de potências aceites.

$$\begin{cases} E_{inf} = 71,83 \cdot (1 - 0,04915) = 68,3W \\ E_{sup} = 71,83 \cdot 1,04915 = 75,4W \end{cases} \quad (4.6)$$

Visto que este intervalo se encontra compreendido entre $[68,3;75,4]W$, é seguro dizer que a medição de $75,3W$ está correta de acordo com o erro relativo presente no módulo de monitorização de potência.

4.1.2 Cenário 2 - Monitorização do funcionamento de um frigorífico combinado

Pretende-se com este cenário de testes, efetuar a monitorização de um frigorífico combinado, com o intuito de verificar a estabilidade do sistema implementado ao monitorizar um maior numero de parâmetros, durante um período de monitorização extenso.

Este equipamento foi escolhido para ser monitorizado devido ao seu tipo de funcionamento termodinâmico e cíclico. Enquanto que o equipamento descrito em 3.1 efetua o aquecimento de uma substância, o frigorífico possui um comportamento inverso retirando calor de duas zonas alvo enviando o mesmo para o exterior.

Em consequência dos cálculos das margens de erro efetuados na secção anterior, é possível considerar os dados obtidos são confiáveis, sendo assim pretende-se também com este cenário de testes estudar a sensibilidade do sistema implementado a estímulos pontuais, sejam estes devido a abertura de portas do frigorífico ou o arranque de um compressor no ciclo de refrigeração.

4.1.2.1 Funcionamento do sistema a monitorizar e dados esperados

O frigorífico que foi escolhido para ser monitorizado é um frigorífico combinado, significando isto que possui um compartimento de refrigeração leve e um compartimento onde a refrigeração é mais intensa, sendo estes o compartimento do frigorífico e do congelador respetivamente.

Adicionalmente, o circuito termodinâmico deste frigorífico é composto por dois ciclos de refrigeração, cada um com um compressor e um evaporador partilhando uma única unidade condensadora de maiores dimensões no exterior, a unidade exterior pode também ser chamada de radiador. A unidade evaporadora ou interior tem como função efetuar a permutação de calor do meio, considerado quente, para o fluido refrigerante fazendo com que este aumente a sua temperatura e evapore dentro das tubagens. Após a receção de calor por parte da unidade evaporadora, o fluido percorre as tubagens com alguma assistência do compressor, ao chegar à unidade condensadora é comprimido com

auxílio de um tubo capilar em junção com o compressor aumentado assim a sua temperatura, o seu ponto de ebulição e o seu estado físico vê-se alterado. Estando no estado líquido a uma temperatura elevada, a permutação de calor para o exterior é efetuada por meio da unidade condensadora ou radiador, fazendo com que o líquido refrigerante diminua a sua temperatura. Visto que, a sua temperatura foi reduzida o líquido encontra-se pronto para ser utilizado de novo nas unidades evaporadoras.

Na figura 4.6 encontra-se o esquema termodinâmico do frigorífico onde é possível identificar as ligações entre os diferentes componentes descritos.

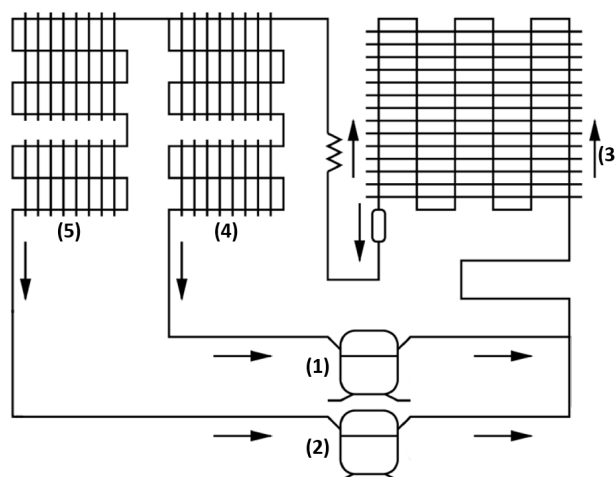


Figura 4.6: Esquema termodinâmico do frigorífico combinado. (1)(2) - Compressores, (3) - Unidade condensadora, (4)(5) - Unidades evaporadoras.

Visto que o sistema termodinâmico é composto por dois ciclos, onde a ativação de um deles é completamente independente do outro, prevê-se que existiram momentos durante o funcionamento do frigorífico e a monitorização do mesmo que exista consumo energético de apenas um dos ciclos, de ambos em simultâneo ou de nenhum.

Posto isto, será possível utilizar os perfis de carga de um frigorífico e congelador convencionais para modelar um perfil de carga teórico para este frigorífico combinado, de acordo com [42, 43] e visto que ambos utilizam como elemento consumidor de energia um compressor, espera-se que os perfis de carga sejam relativamente semelhantes tanto no seu formato como nos picos de arranque, variando apenas no consumo de energia e na sua duração.

Na figura 4.7 encontram-se três perfis de carga, o primeiro perfil refere-se ao frigorífico estando representados três períodos de carga. O segundo perfil é referente ao congelador onde é apresentado apenas um período para o mesmo espaço de tempo, visto que possui um ritmo de funcionamento inferior ao frigorífico. O terceiro perfil trata-se de uma sobreposição dos dois perfis mencionados anteriormente, onde é possível ver três casos possíveis, sem consumo, consumo de apenas uma carga e consumo das duas cargas em simultâneo.

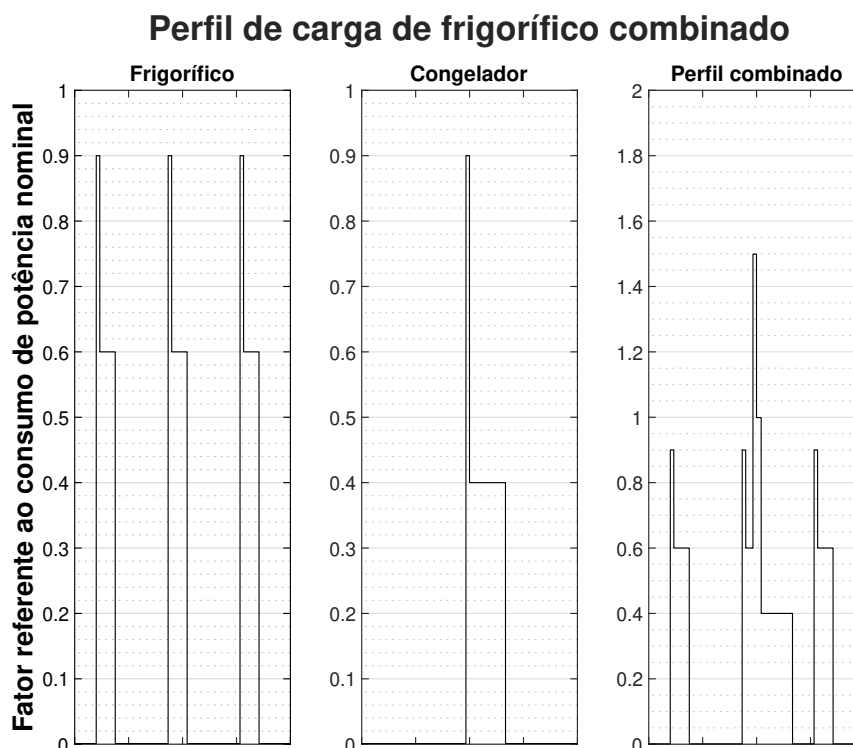


Figura 4.7: Perfil de carga de um frigorífico, congelador e combinados.

4.1.2.2 Sub-cenário 2.1 - Monitorização sem estímulos exteriores.

Neste sub-cenário pretende-se observar o comportamento do frigorífico em condições de funcionamento ideais, ou seja, sem estímulos exteriores como a abertura das portas tanto do frigorífico como do congelador, de maneira a obter uma ideia do seu funcionamento normal em termos energéticos e térmicos.

Para tal, foi utilizado a mesma montagem utilizada no cenário 1, sendo que a única diferença foi a ativação de mais dois sensores de temperatura para que seja possível observar todos os pontos de interesse do frigorífico, sendo estes as temperaturas ambiente, do frigorífico, do congelador e do radiador que se encontra partilhado por ambos os circuitos de refrigeração.

Por inspeção da figura 4.8 é possível observar que ambos os circuitos de refrigeração se encontram ativos, consumindo aproximadamente 80W. Relacionando a temperatura do radiador com as temperaturas internas e o consumo energético é possível determinar que na marca das 1,25 horas o circuito de refrigeração do congelador desligou. Sendo possível chegar a esta conclusão, visto que a temperatura do radiador começa a decrescer de forma gradual enquanto que a temperatura do congelador, apesar de ter alguma latência, é a primeira que começa a aumentar também. Será de notar que como o compartimento

do frigorífico possui uma temperatura interna mais próxima da temperatura ambiente, comparativamente à temperatura do compartimento do congelador, este irá ter um maior impacto no aumento e decréscimo da temperatura do radiador.

Utilizando o mesmo método de inspeção do gráfico é possível confirmar que na marca das 1,5 horas o circuito de refrigeração do frigorífico desliga, sendo também possível confirmar a afirmação anterior de que o circuito de refrigeração do frigorífico tem um maior impacto na temperatura do radiador. Na marca das 2,5 horas é possível determinar que o circuito de refrigeração do congelador voltou a ligar tendo um pico de 705W com um factor de potencia de 0,8 e um período de estabilização relativamente curto visto que a temperatura não se encontrava particularmente elevada. Será também possível observar nas próximas figuras, que este período de estabilização é altamente dependente da temperatura interna dos compartimentos na altura em que os circuito de refrigeração iniciam o seu funcionamento.

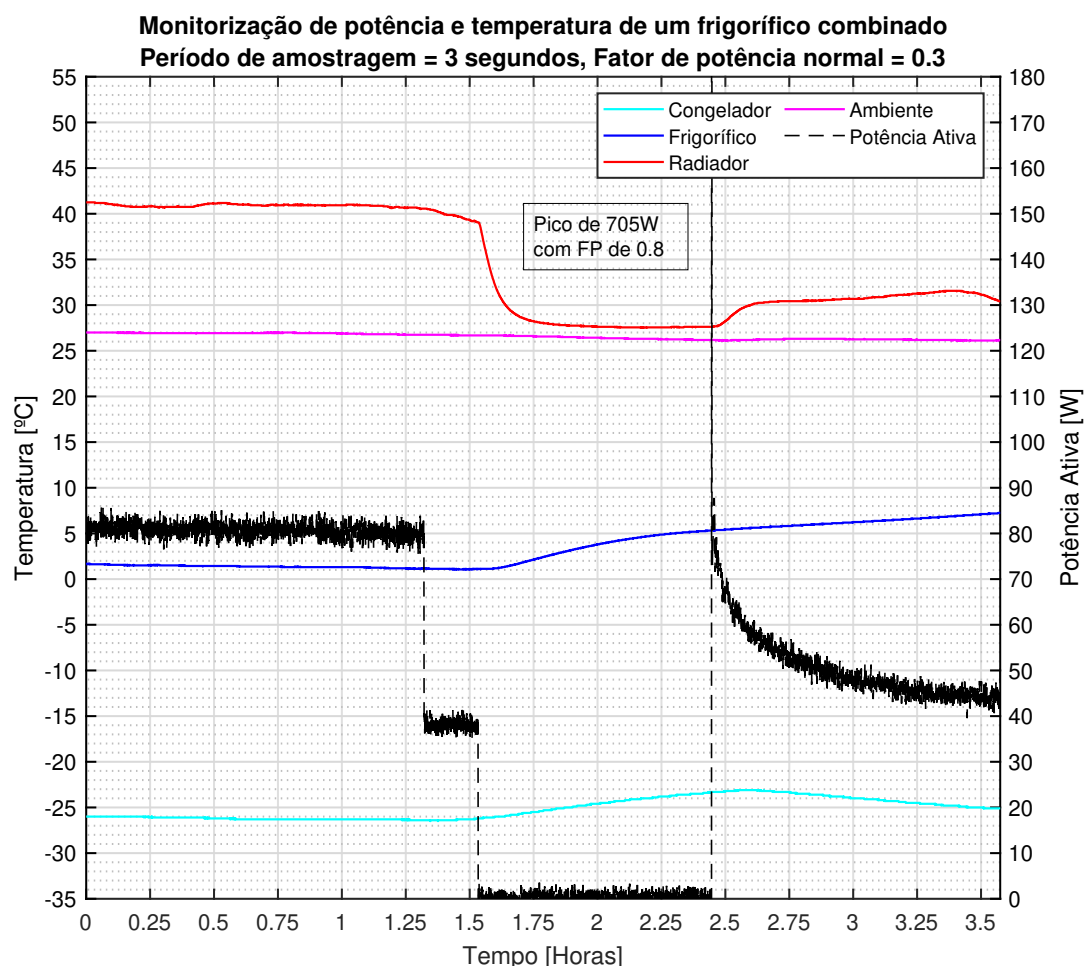


Figura 4.8: Consumo energético e temperatura de um frigorífico combinado em função do tempo. Consumo estável, seguido de um período sem consumo e um arranque do ciclo de refrigeração do congelador.

Na figura 4.9, em baixo, é possível observar o perfil de carga do frigorífico combinado num caso semelhante ao perfil de carga teórico no segundo período da figura 4.7. É possível conferir que não só existem ambos os picos de arranque tanto para o circuito de refrigeração do frigorífico como do congelador, como também existe a sobreposição de carga quando estas se ligam no mesmo espaço de tempo à semelhança do perfil de carga teórico.

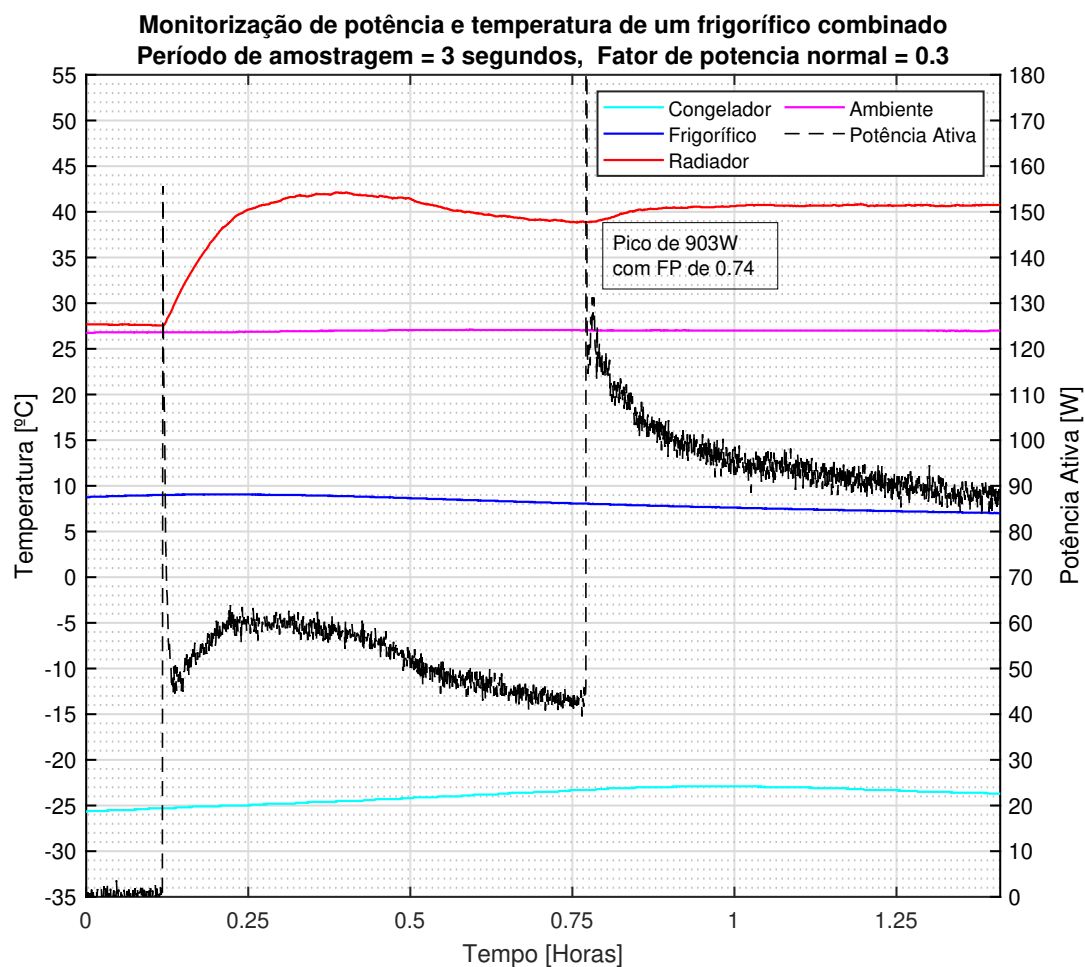


Figura 4.9: Consumo energético e temperatura de um frigorífico combinado em função do tempo. Arranque de ambos os ciclos de refrigeração.

Um dos pontos que é importante referir é facto de que no perfil de carga teórico não existe período de estabilização da carga, no entanto nos dados obtidos na figura 4.9, facilmente se observa que após ligarem os circuitos de refrigeração existe uma pequena sobre-elevação para o circuito de refrigeração do frigorífico e um comportamento de decaimento exponencial para o congelador. Estas diferenças entre os perfis de carga teóricos e reais devem-se ao facto de o frigorífico não ser novo, possuindo aproximadamente 25

anos, fazendo com que exista algum desgaste nos circuitos de refrigeração. Adicionalmente o frigorífico e o congelador encontravam-se com produtos no interior, existia uma massa de gelo no compartimento do congelador e a temperatura ambiente era de aproximadamente 27°C, sendo todos estes fatores que contribuem para uma alta inércia térmica e o aumento de dificuldade para efetuar a permutação de calor, fazendo com que o compressor tenha um maior esforço para efetuar a deslocação e consequente compressão do gás na unidade condensadora resultando assim num maior consumo energético durante este período.

4.1.2.3 Sub-cenário 2.2 - Monitorização com estímulos exteriores

Neste sub-cenário pretende-se observar o comportamento do frigorífico em condições de funcionamento de uso diário ou seja, com a existência de estímulos exteriores como a abertura das portas tanto do frigorífico como do congelador, de maneira a obter alguns resultados práticos sobre a sensibilidade do sistema de monitorização.

Para tal, todas as condições de teste utilizadas no sub-cenário 2.1 foram mantidas, de maneira a criar resultados diretamente comparáveis com o sub-cenário anterior.

No seguimento do teste realizado na sub-secção anterior, figura 4.9, obteve-se os dados presentes na figura 4.10, sendo que esta última é apenas a continuação do teste anterior, onde foi decidido introduzir em uma primeira instância um e apenas um único estímulo exterior, a abertura da porta do frigorífico.

Após o arranque de ambos os circuitos de refrigeração e do período de estabilização das cargas, a porta do frigorífico foi aberta por cerca de 1 minuto, fazendo assim com que a sua temperatura interna aumentasse por 1°C.

Inspecionando o gráfico, presente na figura 4.10, é possível reconhecer uma pequena sobre-elevação tanto da temperatura do frigorífico, como da temperatura do radiador e da potência consumida na marca das 1,75 horas. A sobre-elevação na potência está diretamente relacionada com o aumento da temperatura interior do frigorífico, ou seja a temperatura da unidade evaporadora, por consequência existe uma maior permutação de calor para o fluido refrigerante, fazendo com que o compressor tenha de consumir uma maior quantidade de energia para expelir o calor para o meio ambiente, criando uma sobre-elevação na temperatura do radiador.

Será de notar que existe também um pequeno aumento da temperatura ambiente no entanto, este acontecimento está apenas levemente relacionado com o aumento da temperatura no radiador, pois a real causa do aumento da temperatura ambiente são as atividades do dia-à-dia que estavam a ser realizadas na divisão onde o teste estava a decorrer, como por e.g. a confeção de refeições utilizando alguns eletrodomésticos, nomeadamente o fogão e o forno.

O facto de a temperatura ambiente estar apenas levemente relacionada com a temperatura ambiente, pode ser corroborado, ao relacionar a temperatura ambiente com a

temperatura do radiador no período entre as 0,25 e 0,75 horas, onde é possível observar que um aumento de aproximadamente 10°C no radiador resultou num aumento de apenas 0,5 a 1°C.

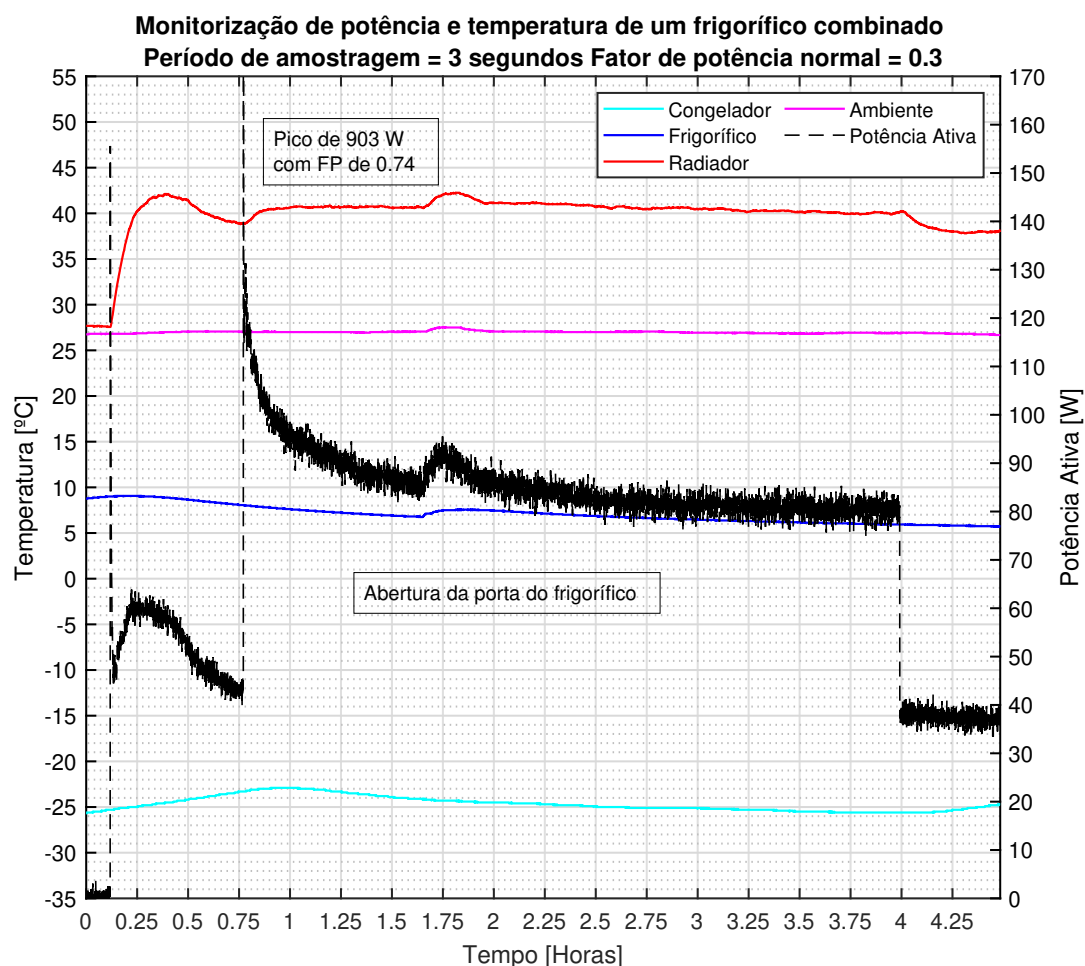


Figura 4.10: Consumo energético e temperatura de um frigorífico combinado em função do tempo. Arranque de ambos os ciclos de refrigeração seguida de uma única abertura da porta do frigorífico.

Possuindo agora os dados necessários, relativos ao comportamento do frigorífico e do sistema de monitorização face a um estímulo exterior, será possível realizar testes mais caóticos onde não se limitam os tempos de abertura das portas nem a sua frequência e ao mesmo tempo manter o nível de controlo e conhecimento necessário para que esses mesmos dados continuem a fazer sentido.

Sendo assim, foi efetuado o teste presente na figura 4.11, este teste além de possuir múltiplas aberturas das portas do frigorífico é também o teste mais longo que é possível efetuar com um período de amostragem de 3 segundos, visto que a plataforma *Azure for IoT* e o serviço *IoT Hub* no seu modelo grátis permitem apenas 8000 mensagens por dia.

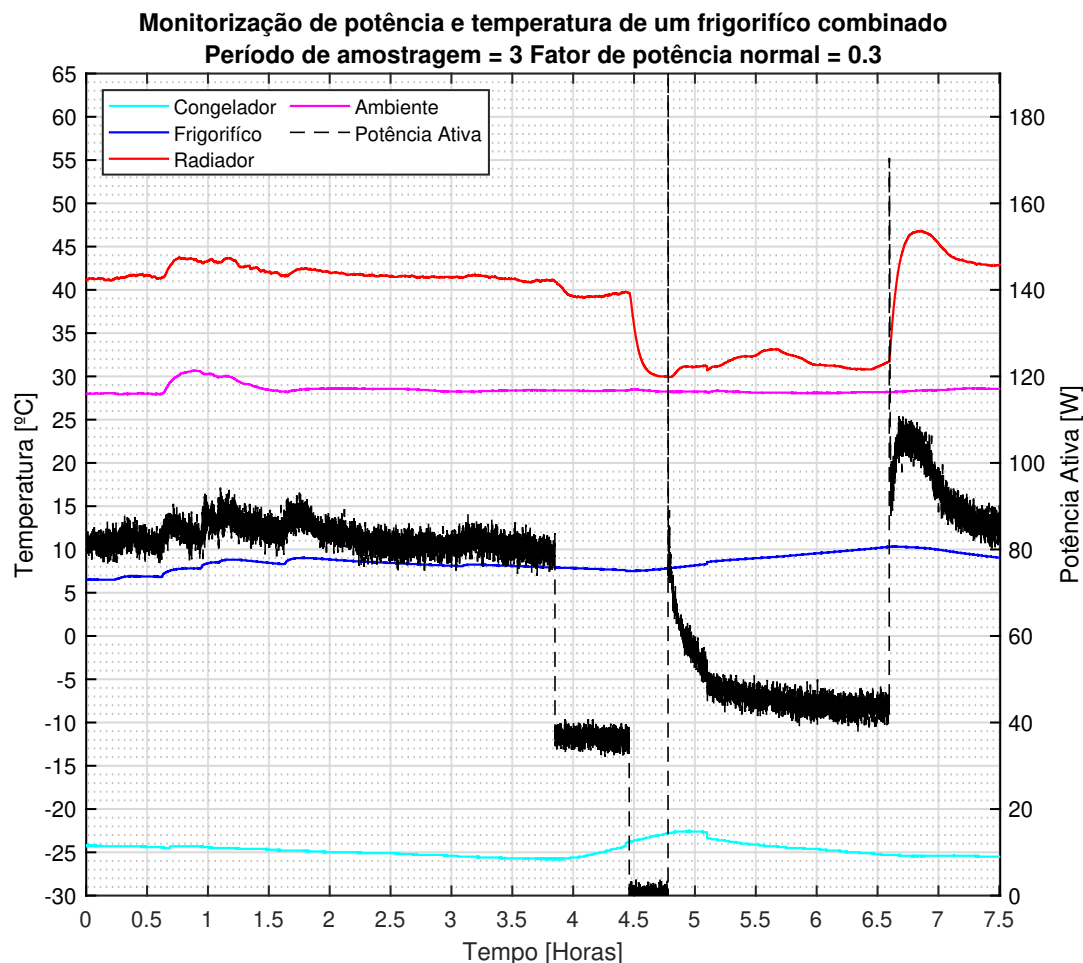


Figura 4.11: Consumo energético e temperatura de um frigorífico combinado em função do tempo. Uso quotidiano.

Inspecionando o gráfico presente na figura 4.11 é possível determinar que nas primeiras duas horas do teste ocorreram 5 aberturas da porta do frigorífico, onde cada uma destas aberturas provoca um comportamento idêntico ao comportamento observado na teste anterior. De modo a despistar falsos positivos todas as aberturas da porta do frigorífico ou do congelador, caso existissem, foram anotados no momento. Posto isto é possível confirmar que as 5 aberturas que se podem observar na gráfico são reais e não existe um único falso positivo.

Ao efetuar este teste, foi também possível determinar que a temperatura a que o circuito de refrigeração do frigorífico desligou foi diferente da temperatura em que o circuito de refrigeração desligou no primeiro teste do sub-cenário 2.1. Sendo que, sem estímulos exteriores o circuito de refrigeração parou o seu funcionamento quando a sua temperatura interna era de 1°C enquanto que neste teste desligou ao chegar a 7°C e voltou a ligar ao chegar aos 10°C como nos teste anteriores. Isto deve-se ao facto de o

termostato encarregue desta função estar embutido nas paredes do frigorífico, causando assim alguma latência e ou insensibilidade a variações rápidas e pontuais de temperatura, como aquelas que estão presentes neste teste.

Posto isto, ao fazer esta sequência de testes em diferentes cenários, foi possível testar a estabilidade do sistema de monitorização ao efetuar testes de longa duração, chegando mesmo a consumir todas as mensagens disponíveis na plataforma. Foi possível também, obter um entendimento geral sobre a sensibilidade do sistema a variações de temperatura e potência sejam estas de grandes ou pequenas magnitudes e do funcionamento interno do equipamento a testar, graças ao facto de ser possível relacionar estas variações com o funcionamento seu teórico e a ocorrência de estímulos não controláveis.

4.1.3 Cenário 3 - Monitorização do funcionamento de um jarro elétrico

O escopo dos cenários anteriores manteve-se maioritariamente sobre o funcionamento, estabilidade e precisão do sistema de monitorização implementado, no entanto, neste terceiro cenário pretende-se efetuar testes num equipamento com um funcionamento semelhante ao equipamento mencionado em 3.1.

Para tal, foi escolhido como alvo dos testes um jarro elétrico, visto que este, à semelhança do equipamento mencionado anteriormente, tem como função o aquecimento de um líquido recorrendo a uma resistência de aquecimento imersa no mesmo, desligando a uma temperatura definida por via de um termostato.

Após escolhido o equipamento a monitorizar é necessário equipar o mesmo com um número adequado de sensores, neste caso como jarro possui a capacidade de aquecer no máximo 1,25 Litros escolheu-se introduzir dois sensores de temperatura no interior do jarro, tendo o cuidado de os por a alturas diferentes com o intuito de confirmar se seria possível verificar os efeitos causados pela convecção do líquido ao aquecer rapidamente e por fim um sensor de temperatura ambiente.

4.1.3.1 Análise dos dados obtidos

Estando o jarro equipado procedeu-se à recolha de dados do funcionamento do mesmo, obtendo assim os gráficos presentes na figura 4.12, visto que se observou alguns comportamentos fora do normal para este tipo de carga foi necessário adicionar dados sobre a tensão, corrente e fator de potência medidos, com o intuito de efetuar alguma resolução de erros. Adicionalmente é possível verificar que na rampa de aquecimento o sensor inferior se encontra mais quente que o sensor superior, indicando que a temperatura da água perto do elemento resistivo é superior à temperatura no topo do jarro.

Analisando os dados obtidos é possível verificar que a potência medida se encontra ligeiramente fora do esperado, sendo o consumo esperado 1900W e o consumo medido 2000W com uma pequena sobre-elevação no início do mesmo. Ao estudar os restantes dados sobre a tensão, corrente e fator de potência é possível verificar que tanto a tensão

como fator de potência medidos possuem comportamentos algo instáveis ao contrário da medição de corrente que se encontra estável em qualquer momento do teste.

Utilizando novamente o medidor PM9C, referido no primeiro cenário, confirmou-se que o valor da tensão apesar de ligeiramente elevado se encontrava estável em qualquer momento do teste, o fator de potência à semelhança da tensão também se encontrava estável e que a corrente medida estava correta, ao contrário dos valores de tensão e fator de potência.

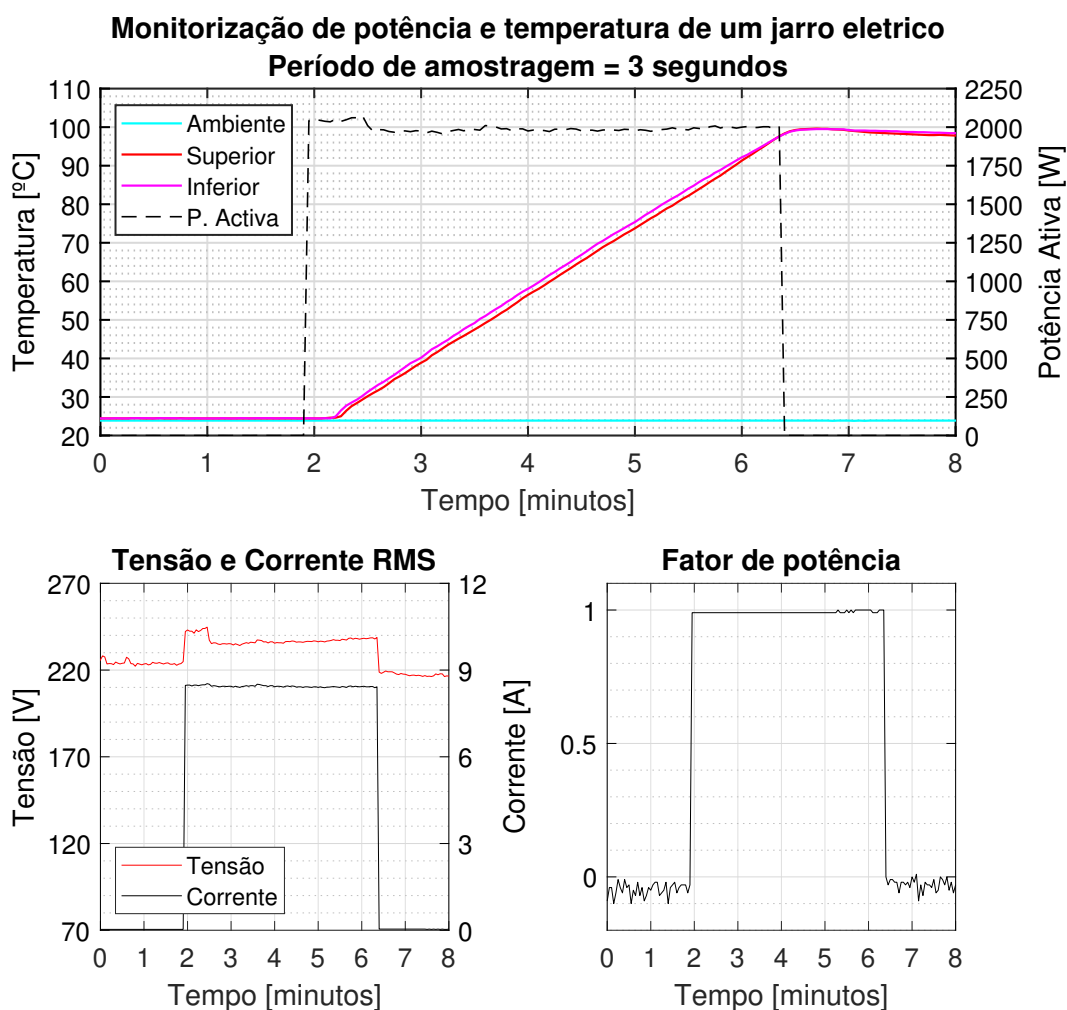


Figura 4.12: Teste de monitorização de um jarro elétrico.

Posto isto, foi possível determinar, ainda que sem certezas, que existe um problema no isolamento do IC ACS71020, fazendo com que a medição de tensão fique incorreta, originando assim medições menores do que o esperado quando a carga não está ligada, e medições demasiado elevadas quando a carga está ativa, estas medições de tensão incorretas são consequentemente utilizadas pelo IC para calcular a potência e o fator de potência, propagando assim o erro.

Visto que existe uma maior facilidade de determinar e modelar exatamente a causa

do problema caso este se intensifique foi decidido efetuar alguns testes a chocolateira industrial descrita em 3.1, com o intuito de verificar se o problema encontrado neste teste persistia com a mesma intensidade ou se aumentava ao utilizar um equipamento de maiores dimensões, suportando consequentemente uma maior massa de água e com um exterior metálico.

4.1.4 Cenário 4 - Monitorização de uma chocolateira industrial

Pretende-se com este cenário de testes, efetuar a monitorização de uma chocolateira industrial de 3000W, figura 4.13, com o intuito de verificar se o problema encontrado no cenário anterior persiste com a mesma intensidade ou se este se agrava, tornando-se assim mais fácil de se detetar e modelar teoricamente de modo a confirmar se é de facto devido à falta de isolamento do IC ACS71020.



Figura 4.13: Chocolateira industrial equipada com sensores de temperatura e monitorizador de potência consumida.

Para tal, a chocolateira foi equipada com três sensores de temperatura em pontos específicos, sendo estes a cuba de aquecimento do chocolate e o compartimento onde foi utilizado PCM, tendo sempre o cuidado de utilizar os sensores e o IC ACS71020 em condições semelhantes ao cenário anterior de maneira a ser possível recriar o problema.

4.1.4.1 Análise dos dados obtidos

Equipada a chocolateira industrial, efetuou-se o aquecimento da mesma até aos 30°C, obtendo assim os dados presentes na figura 4.14. Após uma primeira análise deste teste com a duração de 10 minutos é possível detetar que o formato dos dados obtidos são semelhantes ao do cenário anterior, sendo este o resultado pretendido, onde a tensão medida se encontra com um comportamento fora do normal afetando assim os valores que dependem da mesma, sendo estes o fator de potência e a potência medida.

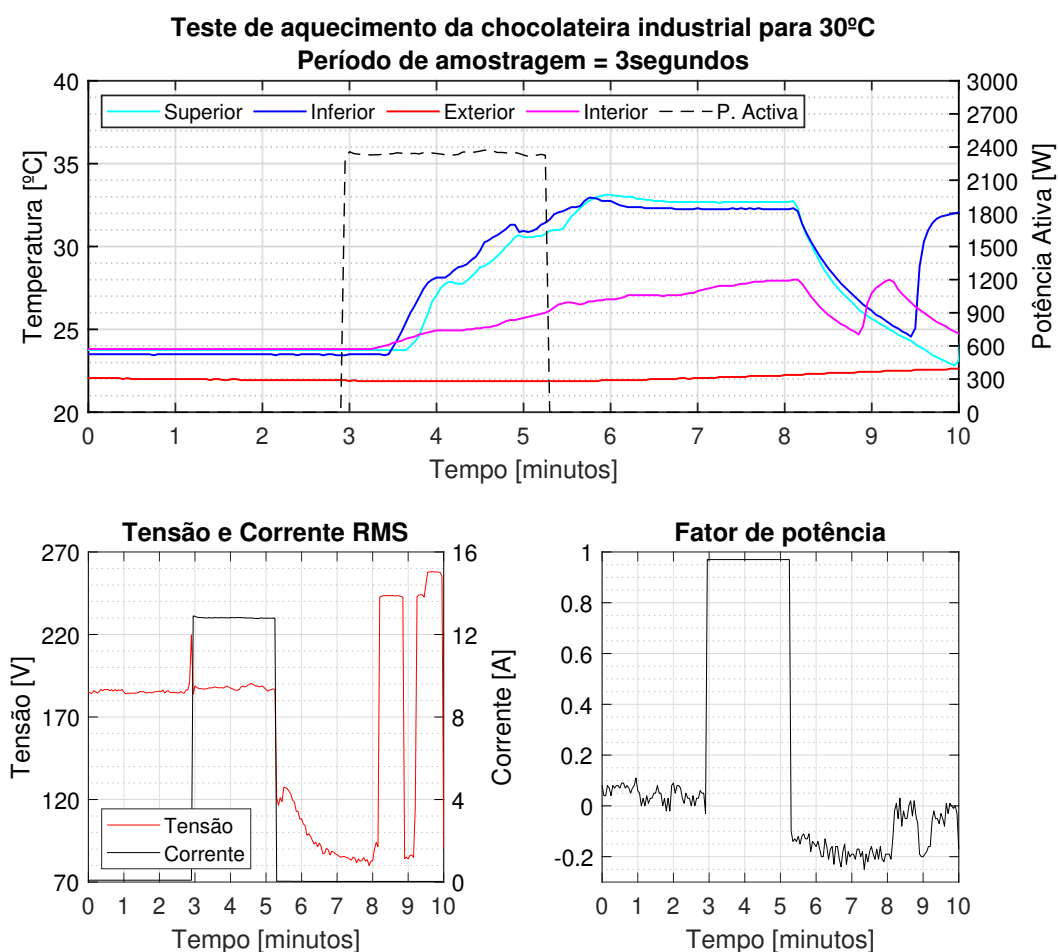


Figura 4.14: Teste de monitorização de uma chocolateira industrial de 3000W.

Ao efetuar uma análise mais detalhada é possível verificar que, a tensão medida antes e durante o período onde a chocolateira está ativa é de aproximadamente 185V, sendo que este valor é consideravelmente menor que o valor de tensão de 225V medido no cenário anterior. Adicionalmente após completado o aquecimento da chocolateira é possível verificar que a tensão medida decresceu para valores próximos de 80V. Comparando estes dados com os dados provenientes do medidor PM9C foi possível determinar que os mesmos se encontram incorretos, sendo o valor real da tensão 231V, confirmando-se assim que o problema se intensificou ao efetuar um teste na chocolateira.

Analisando o problema em questão, assumiu-se que seria devido à falta ou quebra do isolamento do medidor de potência ACS71020 por parte do corpo do sensor de temperatura que estaria avariado, criando assim uma passagem à massa, sabendo que o isolamento é um factor necessário para o bom funcionamento do medidor este problema inviabiliza as medições efetuadas.

A razão para esta teoria é a de que tanto em 4.1.1 como em 4.1.2, onde as medições efetuadas se encontram corretas, o exterior dos equipamentos monitorizados que estão em contacto com os sensores se encontram isolados, sendo que em 4.1.1 o vidro da lâmpada não condutor e em 4.1.2 o corpo do frigorífico ser feito de plástico, outro material não condutor. Enquanto isso, tanto no jarro eléctrico como na chocolateira é plausível que o isolamento seja quebrado devido à existência de uma corrente de fuga, suficientemente pequena para não ser detetada pelo interruptor diferencial, com origem no elemento resistivo que está em contacto com a água presente nos equipamentos e com os sensores que estão imersos na água.

Posto isto, no final do teste os sensores de temperatura foram retirados da água, aproximadamente na marca dos 8 minutos, sendo possível verificar que a temperatura dos mesmos desceu rapidamente e por consequência a tensão medida alterou-se imediatamente para um valor correto, dentro da margem de erro calculada em 4.1.1.2, na marca dos 9 minutos aproximadamente voltou-se a inserir e retirar um dos sensores de dentro de água, fazendo com que a tensão diminui-se para 80V e voltasse para um valor correto, confirmando assim que seria um problema de quebra de isolamento. Após o teste acabado efetuou-se uma medição de resistência entre o exterior metálico dos sensores em questão e a massa conectada ao IC ACS71020 obtendo-se valores entre os 10K Ω e os 27K Ω confirmando assim o problema esperado.

4.1.4.2 Modelação teórica do problema encontrado

Estando a origem do problema determinada, procedeu-se à simulação do sistema utilizado para efetuar a monitorização tanto do jarro eléctrico presente no cenário 3 como da chocolateira industrial presente neste mesmo cenário, com recurso ao software de simulação *LtSpice*.

Na figura 4.15, encontra-se o desenhado o circuito utilizado para efetuar a simulação, sendo possível observar no mesmo todas as ligações efetuadas, desde a rede de baixa tensão conectada a carga e ao divisor resistivo que por sua vez estará ligado ao IC ACS71020 até às correntes de fugas criadas pelo elemento resistivo do equipamento e pelos sensores de temperatura danificados.

De maneira a obter dados úteis para a compreensão do problema em causa, efetuou-se uma simulação, presente na figura 4.16, onde se compara o valor eficaz ou RMS da queda de tensão na resistência R_{sense} com os diferentes valores que a resistência $R11$ e $R5$ podem tomar, sendo estas a ligação à terra e a resistividade entre o corpo do sensor de temperatura e a massa respetivamente.

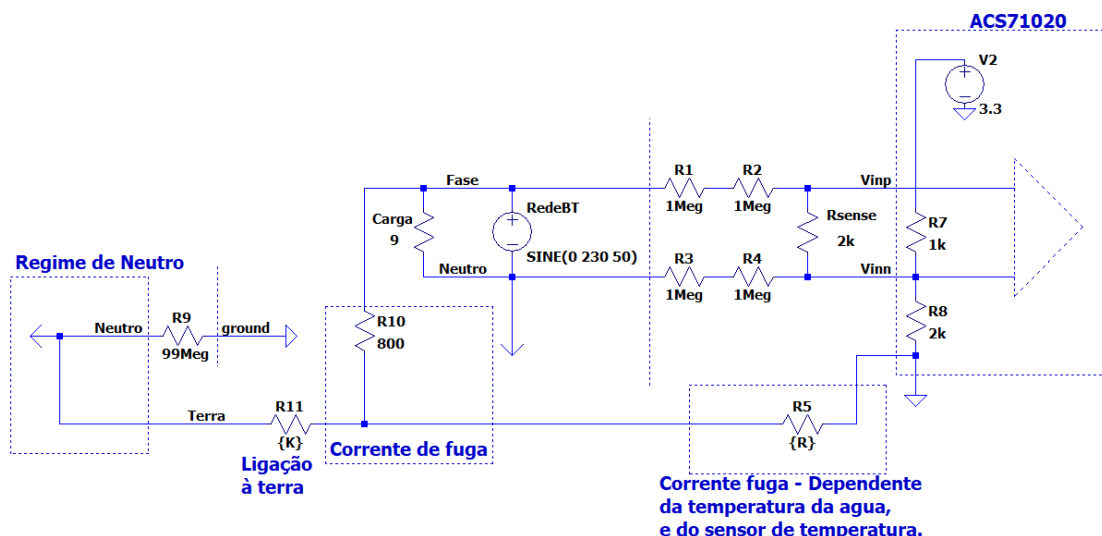


Figura 4.15: Esquema representando o equipamento, sistema de monitorização e problema devido ao isolamento do medidor de potência.

Para a resistência $R11$ foram utilizados dois valores resistivos específicos, com o intuito de simular se o equipamento se encontra conectado ou não à terra de proteção.

Para a resistência $R5$ foram utilizados todos os valores resistivos que esta pode tomar desde 1Ω até $1G\Omega$, significando os mesmo um curto-circuito à massa ou isolamento total respetivamente. Será de notar que ao utilizar este software de simulação é necessário que todos os circuitos possuam como referencia a massa, posto isto de modo a simular o isolamento são utilizadas resistência de valores na ordem dos $M\Omega$ ou $G\Omega$.

Examinando as curvas presentes na figura 4.16, é possível identificar dois comportamentos distintos, sendo estes o aumento da queda de tensão em R_{sense} ou a diminuição, caso a terra de proteção se encontre conectada ou não respetivamente, sendo que o formato das curvas estará diretamente dependente da intensidade da corrente de fuga imposta por $R10$. Esta corrente de fuga dependerá de como o equipamento está construído, no entanto para efeitos de simulação foi utilizado um valor de aproximadamente $288mA$ sendo este um valor plausível e indetetável por um interruptor diferencial de $500mA$ comumente utilizado.

Será de notar que tanto no circuito simulado presente na figura 4.15, como no gráfico da figura 4.16 foram utilizados valores de tensão eficazes com o intuito de simplificar a leitura dos mesmos.

No caso de não se possuir terra de proteção, sendo este o caso que se observou tanto na figura 4.12 e na figura 4.14, a corrente de fuga criada por $R10$ é injetada partir da resistência $R5$ na massa do IC ACS71020, fazendo com que a tensão em V_{inn} aumente, visto que a tensão em V_{inp} se encontra com um valor estático a diferença de potencial entre os dois nós diminui, diminuindo assim a tensão V_{sense} medida pelo IC.

No caso de a terra de proteção estar conectada é possível verificar o comportamento inverso, visto que a tensão em V_{inn} diminui a diferença de potencial entre os dois nós

aumenta, aumentando assim a tensão V_{sense} medida pelo IC chegando mesmo a um ponto onde o amplificador de instrumentação satura corrompendo por completo todas as medições efetuadas.

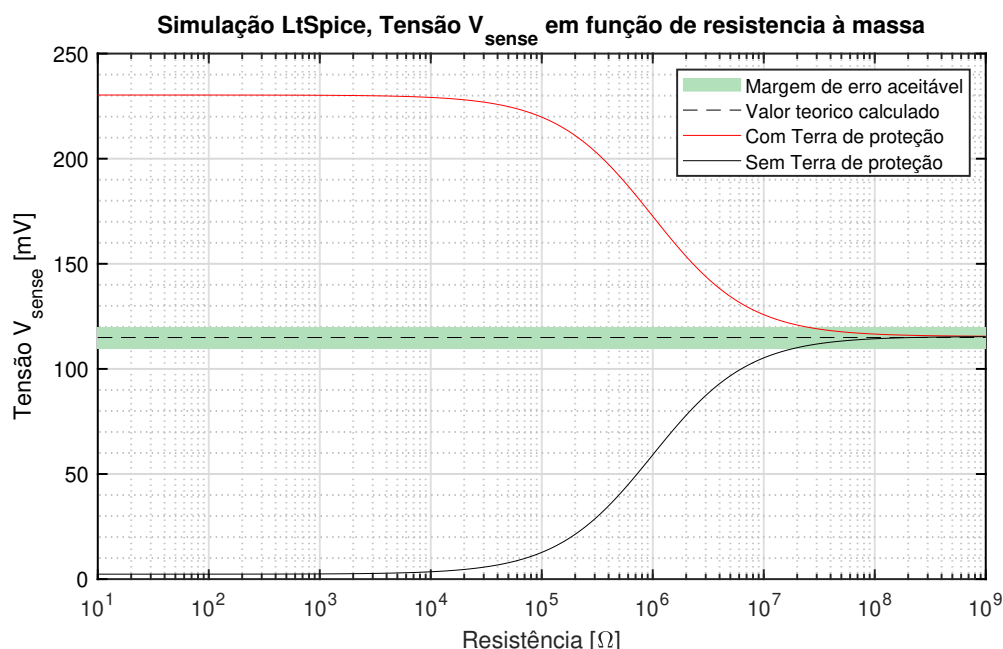


Figura 4.16: Simulação problema de isolamento, queda de tensão eficaz em função da resistência à massa do sensor de temperatura.

Para ambos os casos simulados foi utilizada uma margem de $\pm 5\text{mV}$, identificada a verde, centrada no valor teórico calculado no capítulo anterior, no entanto visto que o circuito interno utilizado pelo IC é extremamente sensível a este tipo de estímulos, não é possível determinar um valor de resistência à massa baixo e próximo o suficiente do valor real onde seja possível obter uma medição de tensão correta.

Apesar de os resultados obtidos neste teste não estarem corretos ao efectuar esta simulação tornou-se óbvio o porquê de o isolamento do IC ACS71020 ser uma característica essencial para o bom funcionamento do mesmo, visto que caso o isolamento seja quebrado é praticamente impossível efectuar uma medição correta com a topologia utilizada.

Posto isto, é possível retificar os erros causados pela medição de tensão, pois a medição de corrente se encontra correta e está imune ao problema de falta de isolamento, adicionalmente como a carga é puramente resistiva o facto de as medições de fator de potência estarem erradas não afetarão o novo calculo de potência, visto que podem também ser corrigidas.

Substituindo o valor de tensão por 231V, valor este medido pelo equipamento de medição PM9C e recalculando os novos valores de potência obtém-se o gráfico seguinte presente na figura 4.17.

No entanto, apesar de ser possível corrigir os dados obtidos é importante mencionar que estes nunca poderão substituir uma medição efetuada corretamente, mesmo assim os

dados corrigidos presentes na figura 4.17 possuem informações relevantes ao funcionamento da chocolateira industrial.

Analisando os dados corrigidos, determina-se que à semelhança do teste efetuado no jarro eléctrico, é possível detetar alguma latência entre o consumo de energia eléctrica e a subida das temperaturas internas dos equipamentos, sendo que devido as dimensões consideravelmente maiores da chocolateira, que resultam em um maior volume de água no seu interior em relação ao jarro eléctrico, obteve-se uma latência térmica superior.

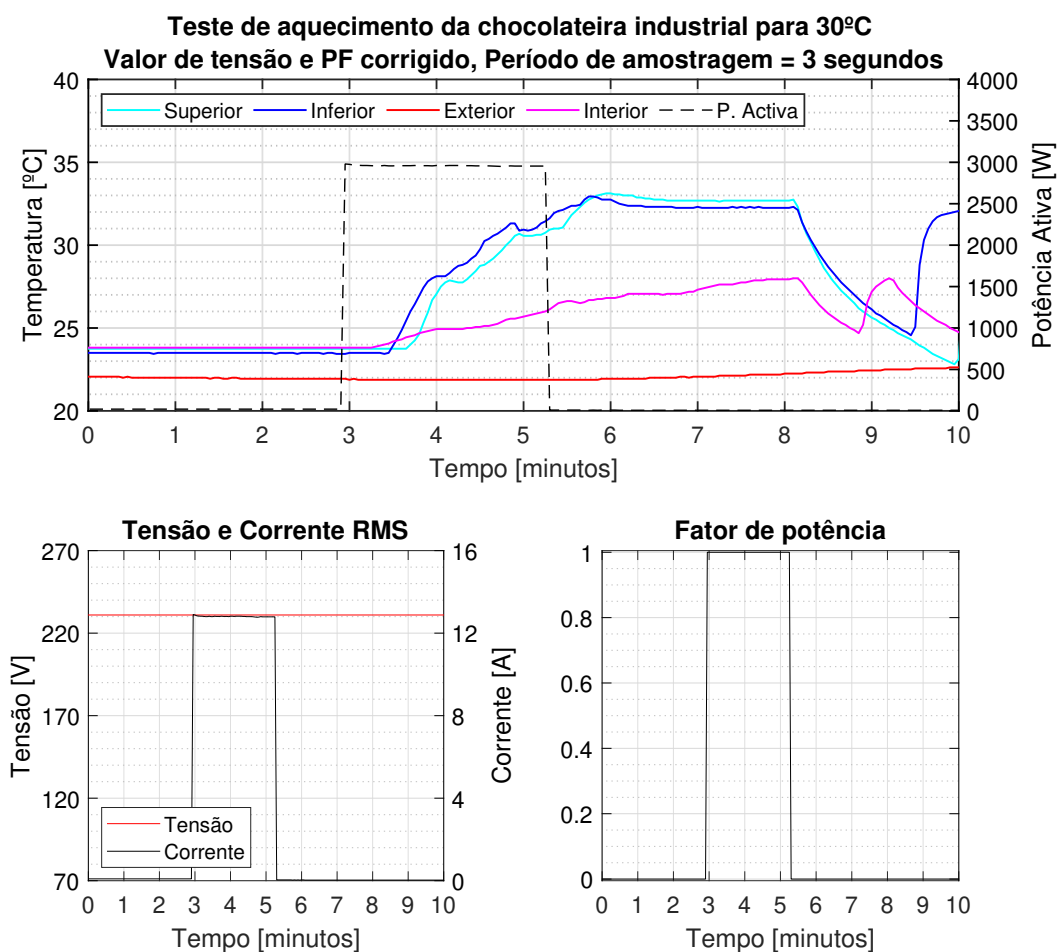


Figura 4.17: Teste de monitorização de uma chocolateira industrial de 3000W com valores de tensão e fator de potência corrigidos.

Outro ponto onde se podem efectuar comparações entre a chocolateira e o jarro é na subida das temperaturas superiores e inferiores do compartimento de aquecimento e no interior do jarro devido ao efeito da convecção. Enquanto que no jarro é possível distinguir um diferencial de 2°C entre a temperatura superior e inferior, na chocolateira devido ao seu porte superior é possível detetar não só uma diferença mais elevada, sendo esta de 4°C em alguns pontos, mas também uma replicação do perfil criado pela subida da temperatura inferior por parte da temperatura superior.

Ao realizar estes diferentes cenários de teste foi possível determinar as limitações atuais do sistema de monitorização implementado, sendo o problema da falta de isolamento catastrófico, pois sem a utilização do medidor PM9C para confirmação os dados obtidos todas as medições efetuadas teriam sido inutilizadas.

Estes resultados, apesar de não serem satisfatórios, pois o sistema de monitorização não funcionou como se pretendia no cenário 3 e 4, são extremamente úteis para o desenvolvimento de uma ferramenta robusta a todo o tipo de ambientes e situações, no entanto é importante não invalidar que o sistema para ambientes menos inóspitos mantém-se funcional com um grau de precisão bastante aceitável. Posto isto, ao saber exatamente os pontos fracos do sistema é possível em uma segunda iteração efectuar a retificação dos mesmos, obtendo assim um sistema de monitorização extremamente robusto.

4.2 Ensaio em equipamento com materiais PCM

Pretendeu-se neste cenário, estudar a capacidade de armazenamento de calor latente dos materiais PCM em aplicações de TES. Para tal, efetuaram-se ensaios com a chocolateira industrial utilizada em 4.1.4, equipada com materiais PCM e uma mistura de chocolate e leite. Esta mistura foi utilizada com a intenção de testar o impacto que os materiais PCM podem ter no processo de aquecimento e arrefecimento controlado de uma dada substância.

Os métodos de TES utilizados na agroindústria, onde é geralmente utilizada água devido à sua facilidade de utilização e ao facto de ser um material inerte, dependem fortemente do armazenamento de calor sensível. Apesar da utilização da água limitar as temperaturas de funcionamento dentro da gama dos 0°C a 100°C, e de fazer com que a temperatura varie facilmente em função da energia armazenada [44]. Fazendo o armazenamento de energia térmica em forma de calor latente, recorrendo a materiais PCM, é possível manter uma temperatura estável durante a maioria do período de armazenamento e, simultaneamente, aumentar a gama de temperaturas de funcionamento.

4.2.1 Casos de estudo e montagem do sistema para o ensaio

De modo a analisar o impacto do PCM, foram considerados quatro casos, envolvendo a carga e descarga do painel de aquecimento de 11,5L com e sem PCM. A cuba de aquecimento do material foi parcialmente cheia, aproximadamente a 60%, com uma mistura de chocolate e leite.

Quanto ao PCM, foi selecionado um material inorgânico composto à base de hidratos de sal, denominado de S34. Este material foi utilizado, pois estes compostos apresentam temperaturas de fusão que se encontram na faixa de operação de processamento de alimentos e capacidades de armazenamento de calor latente superiores aos compostos à base de parafina.

O PCM escolhido (S34) possui uma temperatura de fusão de 34°C, que se adequa à temperatura de fusão da mistura de chocolate e leite. É encapsulado recorrendo a um polímero plástico retangular, dividido em bolsas retangulares de menor dimensão, de modo a diminuir o risco de segregação do sal. Este método de encapsulamento providencia também a flexibilidade mecânica necessária para que o PCM (S34) possa ser aplicado ao painel de aquecimento de uma forma adequada. Na figura 4.18, da esquerda para a direita, observam-se a mistura de chocolate e leite na cuba de aquecimento, o material PCM (S34) onde é possível observar o tipo de encapsulamento utilizado e, por último, o material PCM aplicado ao painel de aquecimento.



Figura 4.18: Esquerda para a direita: mistura de chocolate e leite, PCM (S34) encapsulado e PCM (S34) aplicado na chocolateira.

Foram posicionados 3 sensores de temperatura dentro da chocolateira. Prevendo-se a existência de gradientes de temperatura entre o centro da mistura utilizada e as suas extremidades, os sensores foram aplicados: no painel de aquecimento, de modo a monitorizar o material de aquecimento; na extremidade da mistura de chocolate utilizada; e no centro da mistura. Desta forma, foi possível observar as diferenças de temperatura entre o material presente no painel de aquecimento e os diferentes pontos na mistura. Este posicionamento de sensores permitiu obter informações quanto à inversão das temperaturas no centro da mistura e da sua extremidade no ensaio de controlo sem o uso de PCM no momento da descarga.

Por fim, foi também utilizado em conjunção com os sensores de temperatura o medidor de potência desenvolvido em 3.3.2 e o medidor de potência PM9C.

4.2.2 Análise dos dados obtidos

A partir dos ensaios experimentais, foram obtidos os dados para gerar os gráficos ilustrados nas figuras 4.19 e 4.20. A figura 4.19 refere-se ao ensaio de controlo, onde se utilizou água para aquecer a mistura na cuba e a figura 4.20 refere-se ao ensaio utilizando o material PCM no painel de aquecimento. É também importante de referir que, para o ensaio

de controlo, visto que se utiliza água, o armazenamento térmico é do tipo sensível. Por sua vez, ao utilizar um PCM, o armazenamento é do tipo latente.

Efetuada uma análise comparativa entre casos de ensaio, determinou-se que, com a introdução de um material PCM, apesar do consumo energético ser superior, a capacidade de armazenamento térmico foi melhorada, o ritmo de perda de temperatura foi reduzido, a estabilidade térmica do material foi melhorada e o período de operação foi prolongado.

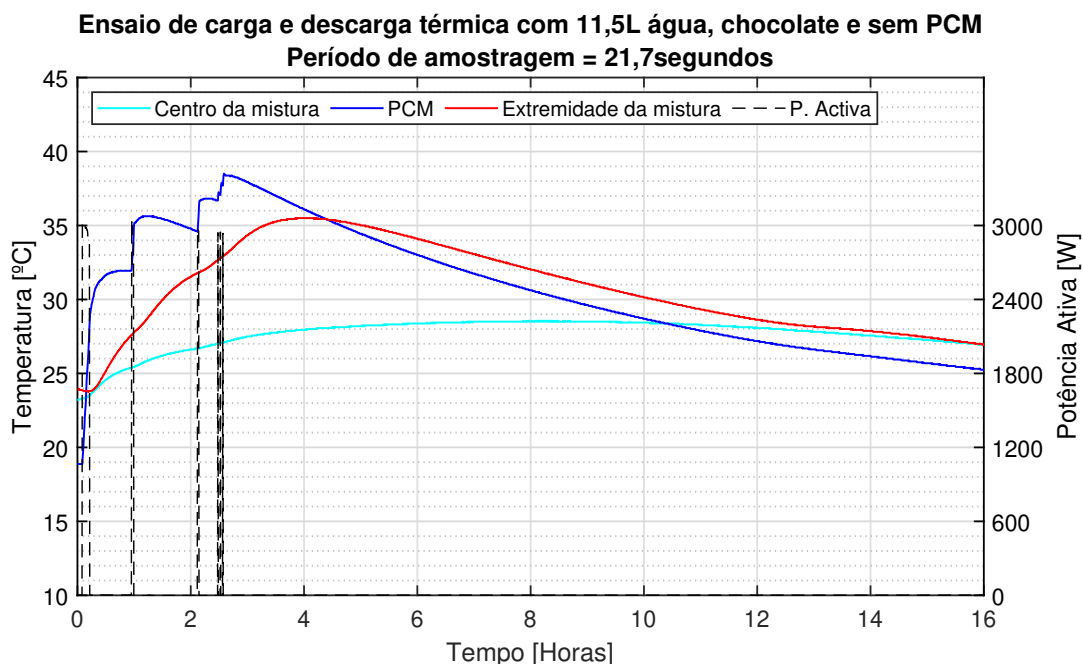


Figura 4.19: Ensaio de controlo utilizando água e uma mistura de chocolate e leite. Carga e descarga com armazenamento térmico sensível.

A solução com PCM (figura 4.20) apresentou um aumento no consumo energético de 54,9% (515Wh para 798Wh) para completar a carga, o qual foi compensado pelo aumento do armazenamento térmico, o prolongamento do ciclo de operação e o aumento da estabilidade térmica, sem a necessidade de reativar o aquecimento. Sendo assim, caso exista possibilidade de alimentar a carga recorrendo a fontes renováveis, em períodos de baixa exploração das mesmas ou em períodos horários onde a tarifa seja menor, será possível obter um ganho de eficiência e flexibilidade energética, não por redução de consumo em si, mas sim pela redução das perdas no seu aproveitamento e consumo.

Na figura 4.20, aquando da descarga foi possível observar que a mistura de chocolate descarrega com aproximadamente o mesmo ritmo do PCM, ao contrário do que acontece na figura 4.19, em que a água arrefece mais rapidamente que a mistura, fazendo com que a própria mistura comece a fornecer energia térmica à água, sendo isto contraproduativo. Como na figura 4.20 não existe o caso onde a mistura fornece energia térmica ao PCM, obtém-se um ganho de aproximadamente 32% (1hora e 20minutos) no período de operação.

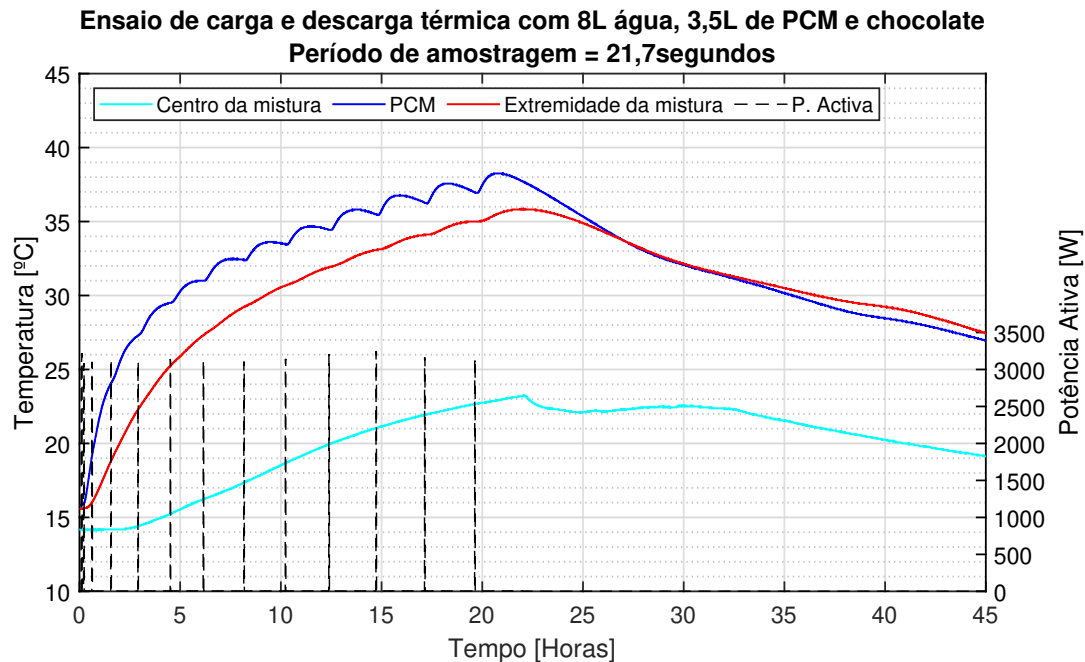


Figura 4.20: Ensaio utilizando PCM e uma mistura de chocolate e leite. Carga e descarga com armazenamento térmico latente.

Além do aumento da capacidade de armazenamento de energia térmica local, a introdução do PCM permite aumentar a estabilidade térmica ao longo do período de operação, visto que o ritmo de perda a abaixo da temperatura de fusão (34°C) foi reduzido de $0,74^{\circ}\text{C/h}$ para $0,46^{\circ}\text{C/h}$.

Por último, observa-se que no processo de descarga, figura 4.19, a variação de temperatura foi linear entre os 38°C e os 30°C . Analogamente, no ensaio utilizando PCM (figura 4.20), observa-se um ponto de inflexão centrado nos 34°C , estando este associado ao início do processo de solidificação do PCM devido à diminuição de energia térmica sensível, tanto por fornecimento à mistura como por perdas de radiação.

No processo destes ensaios, foi demonstrada a capacidade de implementação e incorporação de PCM diretamente na estrutura de equipamentos, substituindo parcialmente a água como fonte de armazenamento de energia térmica, efetuando assim uma passagem de armazenamento térmico latente para sensível, obtendo-se benefícios tanto ao nível do armazenamento térmico como da estabilidade térmica. A substituição da água permitiu um aumento considerável na capacidade de armazenamento de energia térmica.

CONCLUSÃO E TRABALHOS FUTUROS

A adaptação e utilização de diferentes tecnologias e componentes para a implementação de um sistema autónomo com uma determinada função pode apresentar vários desafios na definição das suas funções gerais, na conceção de uma solução prática, no seu custo e por fim na implementação e resolução de problemas.

A utilização de PCM é um exemplo de uma solução com bastante potencial, podendo solucionar alguns problemas referentes à carência energética para o aquecimento de substâncias e à sua ineficiência. No entanto, o uso deste material ainda comporta alguns problemas como praticabilidade e eficácia, visto que é difícil de se determinar o comportamento de um sistema ao ser equipado com este material.

Nesta dissertação foram propostas e desenvolvidas as bases de uma plataforma de monitorização IoT flexível, escalável e de baixo custo com a capacidade de monitorizar sistemas que tenham em algum ponto do seu funcionamento o aquecimento de substâncias. O sistema de monitorização foi dividido em quatro secções que podem ser agrupadas para criar um nó IoT, sendo estas: módulo de sensores, módulo de aquisição de dados, módulo de comunicação e a plataforma IoT. As grandezas a monitorizar que foram escolhidas devido às características predominantes dos PCM's de reter energia térmica e dos equipamentos que têm por base efetuar o aquecimento de uma substância foram inicialmente a potência ativa, o fator de potência e as temperaturas do sistema a monitorizar. No entanto, com a evolução e complexidade do sistema de monitorização foram adicionadas grandezas como tensão e corrente RMS, visto que com estas é possível obter um melhor entendimento de alguns comportamentos inesperados.

Os sensores selecionados adequaram-se às características dos equipamentos a monitorizar, fornecendo dados precisos sobre o funcionamento dos mesmos nos cenários de teste, à exceção dos cenários presentes em 4.1.3 e 4.1.4. Devido ao método de construção dos sensores de temperatura e dado que estes se encontravam imersos num líquido condutor

fizeram com que existissem alguns problemas de falta de isolamento, provocando assim alterações nos dados obtidos provenientes do equipamento responsável pela medição de potência.

O módulo de aquisição de dados encontra-se implementado em conjunto com o nó de comunicação utilizando um protocolo de comunicação simples, fazendo com que alterações aos dados transmitidos sejam efetuadas com facilidade para que este se adeque a qualquer tipo de combinação de sensores e dados adquiridos.

A plataforma IoT escolhida, encontra-se como uma das melhores não só no seu desempenho e características mas também relativamente ao seu custo, a implementação de funções futuras encontra-se também simplificada devido à documentação detalhada que é fornecida aos utilizadores.

Após efetuada a validação da plataforma foi realizado um ensaio onde se testou o funcionamento de um dado equipamento, antes e depois de ser introduzido o material PCM. Neste ensaio foi possível determinar que, ao utilizar PCM para efetuar o armazenamento de energia térmica latente o ciclo de operação e a estabilidade térmica do equipamento eram melhorados sem que existisse a necessidade de reativar o elemento de aquecimento, no entanto, o consumo energético deste mesmo equipamento aumentou em 54,9%. Sendo assim, caso exista possibilidade de alimentar a carga recorrendo a fontes renováveis, em períodos de baixa exploração das mesmas ou em períodos horários onde a tarifa seja menor, será possível obter um ganho de eficiência e flexibilidade energética.

Posto isto, o nó IoT implementado cumpre todas as especificações impostas, fornecendo armazenamento de dados e um método de monitorização dos mesmos em tempo real com uma precisão notável, exceto em algumas condições pontuais. Este sistema de monitorização devido às métricas que monitoriza, possui a capacidade de se tornar uma ferramenta útil no controlo e estudo da implementação de PCM para o armazenamento térmico latente.

5.1 Trabalhos futuros

Primeiramente, sugere-se a adição de um maior conjunto de sensores prontos a utilizar em conjunto com o módulo de aquisição de dados, fazendo efetivamente com que exista uma maior facilidade de se monitorizar diferentes equipamentos consoante a necessidade. Adicionalmente, a criação de bibliotecas para os diversos sensores é visto como um ponto importante no que toca à rápida implementação dos mesmos.

Em segundo lugar, propõe-se a alteração da implementação do IC ACS71020. A adição de um método de derivação de alimentação elétrica vinda da rede de baixa tensão, utilizando um pequeno transformador redutor com um retificador de onda completa e um conversor em modo buck em conjunto com algum mecanismo de isolamento galvânico das linhas de comunicação é imperativo para o bom funcionamento do IC em todas as situações de funcionamento por mais adversas que estas se encontrem.

Depois, sugere-se a alteração dos sensores de temperatura a utilizar caso estes necessitem de estar imersos em líquidos condutores. Visto que com o aquecimento e arrefecimento sucessivo dos sensores pode surgir falhas no invólucro, criando assim situações problemáticas difíceis de processar e descobrir, tornando uma implementação simples e rápida num processo moroso e ineficiente.

Por último, sugere-se a utilização de algumas funções presentes na plataforma *Azure for IoT* de maneira a acelerar e melhorar o desempenho da mesma. A utilização do mecanismo de adição automática de dispositivos IoT ao serviço *IoT Hub* é vista como uma boa adição ao melhoramento e facilidade de implementação dos sistemas de monitorização. A adição de serviços de inteligência artificial e *machine learning* poderão também, trazer mais-valias ao sistema de monitorização, visto que estes podem ajudar na descoberta de padrões de utilização, fenómenos difíceis de relacionar e na deteção preventiva de problemas e falhas.

BIBLIOGRAFIA

- [1] P. Linares e X. Labandeira. “Energy efficiency: Economics and policy”. Em: *Journal of Economic Surveys* 24.3 (2010), pp. 573–592. ISSN: 09500804. DOI: 10.1111/j.1467-6419.2009.00609.x.
- [2] D. Y. Perraudin, S. R. Binder, E. Rezaei, A. Ortona e S. Haussener. “Phase change material systems for high temperature heat storage”. Em: *Chimia* 69.12 (2015), pp. 780–783. ISSN: 00094293. DOI: 10.2533/chimia.2015.780.
- [3] A. Bejan. *Thermodynamics Other books by Adrian Bejan* : ISBN: 9781119052098.
- [4] P. P. Urone. *Heat*. 2019. URL: [https://phys.libretexts.org/Bookshelves/College_Physics/Book%3A_College_Physics_\(OpenStax\)/14%3A_Heat_and_Heat_Transfer_Methods/14.02%3A_Heat](https://phys.libretexts.org/Bookshelves/College_Physics/Book%3A_College_Physics_(OpenStax)/14%3A_Heat_and_Heat_Transfer_Methods/14.02%3A_Heat).
- [5] A. S. L. Frank P. Incropera, David P. Dewitt, Theodore L. Bergman. *Fundamentals of Heat and Mass Transfer*. Vol. 53. 9. 2013, pp. 1689–1699. ISBN: 9788578110796. DOI: 10.1017/CB09781107415324.004. arXiv: arXiv:1011.1669v3.
- [6] H. Mehling e L. F. Cabeza. “PHASE CHANGE MATERIALS AND THEIR BASIC PROPERTIES”. Em: *Thermal Energy Storage for Sustainable Energy Consumption*. Ed. por H. Ö. Paksoy. Dordrecht: Springer Netherlands, 2007, pp. 257–277. ISBN: 978-1-4020-5290-3.
- [7] D. Lora. “Phase change material product design. Market and business development assessment in the food industry”. Em: *Instituto superior técnico de lisboa* (2014), p. 113.
- [8] S. Pereira e L. Costa. “Armazenamento de Energia Térmica Através de Materiais de Mudança de Fase”. Em: *Faculdade de engenharia universidade do porto* (2014), p. 162. URL: <https://repositorio-aberto.up.pt/bitstream/10216/73754/2/98207.pdf>.
- [9] N. Pfleger, T. Bauer, C. Martin, M. Eck e A. Wörner. “Thermal energy storage - overview and specific insight into nitrate salts for sensible and latent heat storage”. Em: *Beilstein Journal of Nanotechnology* 6.1 (2015), pp. 1487–1497. ISSN: 21904286. DOI: 10.3762/bjnano.6.154.

- [10] D. Zhou. "Review on thermal energy storage with phase change materials (PCMs) in building applications". Em: *School of Engineering, University of Warwick; 2School of Mechanical Engineering, Shanghai Jiaotong University* (2012), pp. 593–605.
- [11] K. Pielichowska e K. Pielichowski. "Phase change materials for thermal energy storage". Em: *Progress in Materials Science* 65.May 2012 (2014), pp. 67–123. ISSN: 00796425. DOI: 10.1016/j.pmatsci.2014.03.005.
- [12] S. Mondal. "Phase change materials for smart textiles - An overview". Em: *Applied Thermal Engineering* 28.11-12 (2008), pp. 1536–1550. ISSN: 13594311. DOI: 10.1016/j.applthermaleng.2007.08.009.
- [13] C. Sunliang. "State of the Art Thermal Energy Storage Solutions for High Performance Buildings". Em: *University of Jyväskylä* (2010), p. 201.
- [14] A. Sharma, V. V. Tyagi, C. R. Chen e D. Buddhi. "Review on thermal energy storage with phase change materials and applications". Em: *Renewable and Sustainable Energy Reviews* 13.2 (2009), pp. 318–345. ISSN: 13640321. DOI: 10.1016/j.rser.2007.10.005.
- [15] N. Sarier e E. Onder. "Organic phase change materials and their textile applications: An overview". Em: *Thermochimica Acta* 540 (2012), pp. 7–60. ISSN: 00406031. DOI: 10.1016/j.tca.2012.04.013. URL: <http://dx.doi.org/10.1016/j.tca.2012.04.013>.
- [16] G. A. Lane. *Solar Heat Storage: Latent Heat Materials Vol1*. 1983.
- [17] S. Sharma e K. Sagara. "Latent Heat Storage Materials and Systems: A Review". Em: *International Journal of Green Energy* 2.1 (2005), pp. 1–56. ISSN: 1543-5075. DOI: 10.1081/ge-200051299.
- [18] D. W. Hawes, D. Feldman e D. Banu. "Latent heat storage in building materials". Em: *Energy and Buildings* 20.1 (1993), pp. 77–86. ISSN: 03787788. DOI: 10.1016/0378-7788(93)90040-2.
- [19] H. Ö. Paskoy. *Thermal energy Storage for sustainable Energy Consumption, Fundamentals, Case Studies and Design*. ISBN: 9781402052880.
- [20] R. van Kranenburg e A. Bassi. "IoT Challenges". Em: *Communications in Mobile Computing* 1.1 (2012), pp. 1–5. ISSN: 2192-1121. DOI: 10.1186/2192-1121-1-9.
- [21] L. Atzori, A. Iera e G. Morabito. "The Internet of Things: A survey". Em: *Computer Networks* 54.15 (2010), pp. 2787–2805. ISSN: 13891286. DOI: 10.1016/j.comnet.2010.05.010.
- [22] J. H. Nord, A. Koohang e J. Paliszkievicz. "The Internet of Things: Review and theoretical framework". Em: *Expert Systems with Applications* 133 (2019), pp. 97–108. ISSN: 09574174. DOI: 10.1016/j.eswa.2019.05.014.
- [23] Espressif. *ESP32 Overview | Espressif Systems*. 2017. URL: <https://www.espressif.com/en/products/hardware/esp32/overview>.

-
- [24] E. Embedded. *Overview of ESP32 features*. URL: https://www.exploreembedded.com/wiki/Overview_of_ESP32_features._What_do_they_practically_mean%3F.
 - [25] CYPRESS PSOC. *32-bit Arm® Cortex®-M3 PSoC® 5LP*. URL: <https://www.cypress.com/products/32-bit-arm-cortex-m3-psoc-5lp>.
 - [26] Raspberry Pi Foundation. *Raspberry Pi 3 Model B+ Datasheet*. 2016. URL: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>.
 - [27] Atmel. *ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET*. 2015.
 - [28] Espressif. *ESP32 Series Datasheet*. 2019. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
 - [29] C. C. Liu, S. J. Chang, G. Y. Huang e Y. Z. Lin. "A 10-bit 50-MS/s SAR ADC with a monotonic capacitor switching procedure". Em: *IEEE Journal of Solid-State Circuits* 45.4 (2010), pp. 731–740. ISSN: 00189200. DOI: 10.1109/JSSC.2010.2042254.
 - [30] M. R. Miller e C. S. Petrie. "A multibit sigma-delta ADC for multimode receivers". Em: *IEEE Journal of Solid-State Circuits* 38.3 (2003), pp. 475–482. ISSN: 00189200. DOI: 10.1109/JSSC.2002.808321.
 - [31] Maximintegrated. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer DS18B20 Programmable Resolution 1-Wire Digital Thermometer Absolute Maximum Ratings*. 2019. URL: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
 - [32] R. M. Solutions. *Monnit Wireless temperature sensor datasheet*. 2020. URL: <https://monnit.azureedge.net/content/documents/datasheets/sensors/MDS-001-Temperature-Sensor-Data-Sheet.pdf>.
 - [33] Fischer. *NC56 Capacitive Level Sensor - Datasheet*. 2000. URL: http://www.alvi.com.au/wp-content/product-pdf/DB_GB_NC56.pdf.
 - [34] Texas Instruments. *MSP430AFE2x3 single-phase energy meter IC*. 2011. URL: https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/196/slyt428.pdf.
 - [35] M. Integrated. *AC Power Measurement IC Table of Contents*. 2012. URL: <https://datasheets.maximintegrated.com/en/ds/78M6613.pdf>.
 - [36] A. Microsystems. *Single Phase , Isolated , Power Monitoring IC with Voltage Zero Crossing and Overcurrent Detection*. 2018. URL: <https://www.allegromicro.com/en/products/sense/current-sensor-ics/zero-to-fifty-amp-integrated-conductor-sensor-ics/acs71020>.
 - [37] Campeona. *Txm/20-lb Datasheet*. URL: <http://campeona.es/product.php?lang=en¶m1=txm20-lb>.

- [38] Cypress. *DS18x8 Digital Temperature Sensor General description Input-output connections*.
- [39] A. Microsystems. *User Manual User Manual*. 2012. URL: <https://www.allegromicro.com/~media/Files/Demo-Boards/ASEK71020-Eval-Board-User-Manual.ashx>.
- [40] H. Hejazi, H. Rajab, T. Cinkler e L. Lengyel. "Survey of platforms for massive IoT". Em: *2018 IEEE International Conference on Future IoT Technologies, Future IoT 2018* 2018-Janua (2018), pp. 1–8. DOI: 10.1109/FIOT.2018.8325598.
- [41] P. Pierleoni, R. Concetti, A. Belli e L. Palma. "Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison". Em: *IEEE Access* 8 (2020), pp. 5455–5470. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2961511.
- [42] R. Stamminger. "Synergy Potential of Smart Appliances". Em: *D2.3 of WP2 from the Smart-A project* (2008), p. 237. ISSN: 1540-7977. URL: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5452928.
- [43] M. Pipattanasomporn, M. Kuzlu, S. Rahman e Y. Teklu. "Load profiles of selected major household appliances and their demand response opportunities". Em: *IEEE Transactions on Smart Grid* 5.2 (2014), pp. 742–750. ISSN: 19493053. DOI: 10.1109/TSG.2013.2268664.
- [44] S. Furbo. *Using water for heat storage in thermal energy storage (TES) systems*. 2015, pp. 31–47. ISBN: 9781782420965. DOI: 10.1533/9781782420965.1.31.



CÓDIGO UTILIZADO NO MCU PSoC 5LP

Listagem I.1: Código PSoC 5Lp.

```

1  /* =====
2  /*
3  /* Autor: Diogo Alexandre Ferreira,
4  /* Ano: 2020, Novembro
5  /* Descricao: Código pertencente ao Nó IoT
6  /* com o intuito de fazer uma monitorização
7  /* integrada de materiais PCM.
8  /*
9  /* =====
10 */
11 #include "project.h"
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <math.h>
15
16 #define high 1
17 #define low 0
18
19 // *****Endereços ACS71020*****
20 #define VrmsIrms 0x20
21 #define Pactive 0x21
22 #define Saparent 0x22
23 #define Qreactive 0x23
24 #define PFactor 0x24
25 #define Pinst 0x2c
26 #define Numsamp 0x25
27
28 #define ACCESS 0x2e
29 #define code 0x4F70656E

```

```

30     #define read 0x80
31     #define write 0x00
32 //*****Variaveis Globais*****
33     float VRMs = 0.0000;    //Tensao RMS
34     float IRMs = 0.0000;    //corrente RMS
35     float P = 0.0000;       //Potência activa
36     float S = 0.0000;       //Potência aparente
37     float Q = 0.0000;       //Potência Reactiva
38     float PF = 0.0000;      //Factor de potência
39 //*****Constantes Globais*****
40     float fullscaleV =550.0; //Full scale voltage
41     float IPR = 30.0;        //Dispositivo vem calibrado de fábrica
42     float Vstep= 0.00003051757; //passar para numero fraccionário 15bitfrac
43     float Istep= 0.00006103515; //passar para numero fraccionário 14bitfrac
44     float PFstep=0.001953125; //passar para numero fraccionário 9bitfrac
45     float MxPwr=16500.0;     //full scale power calculated by fullscaleV*IPR
46     int TRIM = 0;           //Trims readings to zero when voltage is under 15volt
47 //*****Variaveis globais de temperatura*****
48     float T1 = 0;
49     float T2 = 0;
50     float T3 = 0;
51     float T4 = 0;
52
53
54 void SPIrequestData32 (int rw, uint8 addr, uint8 p, uint8 s, uint8 t, uint8 q){
55 //*****
56 /* Pedido de dados de um registo com permissoes de leitura ou escrita
57 /* e guarda uma resposta de 32btis no buffer Rx de SPI
58 /* Inputs int RW, uint8 addr Clears TxRx&FIFO
59 /* RW == 1 READ RW==0 Write
60 /* resposta de 32bit é guardada em SPI_RxDataBuffer
61 //*****/
62     uint8 Address = 0x00;
63     //clear buffers
64     SPIM_1_ClearTxBuffer();
65     SPIM_1_ClearRxBuffer();
66     SPIM_1_ClearFIFO();
67     //enviar o enderesso com o bit de RW
68     //uint8 Address = ((0x1E) | read);
69     if(rw==1){
70         Address = ((addr) | read);
71     }else if (rw == 0){
72         Address = ((addr) | write);
73     }
74     //esperar uma resposta de 32bit == 4 frames
75     if(rw==1){
76         SPIM_1_WriteTxData(Address);
77         SPIM_1_WriteTxData(0x00);
78         SPIM_1_WriteTxData(0x00);
79         SPIM_1_WriteTxData(0x00);

```

```

80     SPIM_1_WriteTxData(0x00);}
81     if(rw==0){
82     SPIM_1_WriteTxData(Address);
83     SPIM_1_WriteTxData(p);
84     SPIM_1_WriteTxData(s);
85     SPIM_1_WriteTxData(t);
86     SPIM_1_WriteTxData(q);}
87     while(!(SPIM_1_ReadTxStatus() & SPIM_1_STS_SPI_DONE));
88
89 }
90
91 void resetvars(){
92 /*****
93 /*   Reset a variaveis globais ( VRMs IRMs P S Q and PF)
94 /*****/
95     VRMs = 0.0000;    //Tensao RMS
96     IRMs = 0.0000;    //corrente RMS
97     P = 0.0000;       //Potência activa
98     S = 0.0000;       //Potência aparente
99     Q = 0.0000;       //Potência Reactiva
100    PF = 0.0000;       //Factor de potência
101
102 }
103
104 void V_I_RMS (){
105 /*****
106 /*   Transforma uma resposta de 32bits em VRMS(V) and IRMS(A) values
107 /*   output variaveis globais VRMs IRMs
108 /*****/
109     uint8 primeiro = 0;
110     uint8 segundo = 0;
111     uint8 terceiro = 0;
112     uint8 quarto = 0;
113     uint16 V = 0;
114     uint16 I = 0;
115     uint16 aux = 0;
116     uint16 aux2 = 0;
117     for(uint8 i = 0u; i<5u; i++){
118         //ignorar o primeiro
119         switch(i){
120             case 0: SPIM_1_ReadRxData(); break;    //ingorar o primeiro
121             case 1: primeiro = SPIM_1_ReadRxData(); break;
122             case 2: segundo = SPIM_1_ReadRxData(); break;
123             case 3: terceiro = SPIM_1_ReadRxData(); break;
124             case 4: quarto = SPIM_1_ReadRxData(); break;
125             default:break;
126         }
127
128     }
129     //Tensao é um numero fraccionario de 15 bits, 16obit

```

```

130     V = ((segundo<<8) | primeiro);
131     VRMs = (fullscaleV *( (float)V / (float)0x8000) );
132
133     //corrente é um numero de 15bits com 14bits fraccionários
134     I = ((quarto<<8) | terceiro);
135     //mascara para tirar o numero das unidades
136     aux = I & 0b0100000000000000;
137     if(aux == 0x4000){ aux = 1;}
138     else if (aux == 0){aux = 0;}
139     //mascara para retirar o numero fraccionário
140     aux2 = (I & 0b0011111111111111);
141     //corrente é dada pela multiplicação do IPR pela soma
142     //das unidades com a parte fraccionária
143     IRMs = IPR*(aux + (aux2*Istep));
144
145
146 }
147
148 void ActivePower (){
149     /*****
150     /* Transforma uma resposta de 32bits em potencia in Watts
151     /* output variaveis globais P
152     *****/
153     uint8 primeiro = 0;
154     uint8 segundo = 0;
155     uint8 terceiro = 0;
156     uint16 Power = 0;
157     float neg=0.0;
158     uint16 aux=0;
159     uint16 unid=0;
160     uint16 frac=0;
161     for(uint8 i = 0u; i<5u; i++){
162         //ignorar o primeiro
163         switch(i){
164             case 0: SPIM_1_ReadRxData(); break; //ingorar o primeiro
165             case 1: primeiro = SPIM_1_ReadRxData(); break;
166             case 2: segundo = SPIM_1_ReadRxData(); break;
167             case 3: terceiro = SPIM_1_ReadRxData(); break;
168             default:break;
169         }
170
171     }
172     //potencia activa é um signed17bit com 15bits fraccionários
173     Power = ((segundo<<8) | primeiro);
174     terceiro = terceiro & 0b00000001; //retirar o bit de sinal
175     if( terceiro == 1 ){neg = -1.0; Power = ~Power;}
176     else if( terceiro == 0){neg = 1.0;}
177
178
179     aux = Power & (0b1000000000000000); //retirar o bit das unidades

```

```

180
181     if( aux == 0x8000){unid = 1.0;}
182     else if(aux == 0){unid = 0.0;}
183
184     frac= Power & (0b0111111111111111);    //retirar a parte fraccionaria
185
186     P=MxPwr*neg*(unid + ((float)frac/(float)0x8000) );
187
188
189 }
190
191 void ApparentPower (){
192     /*****
193     /* Transforma uma resposta de 32bits em VoltAmpere
194     /* output variaveis globais S
195     /*****/
196     uint8 primeiro = 0;
197     uint8 segundo = 0;
198     uint16 Power = 0;
199     uint16 aux=0;
200     uint16 unid=0;
201     uint16 frac=0;
202     for(uint8 i = 0u; i<5u; i++){
203         //ignorar o primeiro
204         switch(i){
205             case 0: SPIM_1_ReadRxData(); break;    //ingorar o primeiro
206             case 1: primeiro = SPIM_1_ReadRxData(); break;
207             case 2: segundo = SPIM_1_ReadRxData(); break;
208             default:break;
209         }
210     }
211     Power = ((segundo<<8) | primeiro);
212
213     aux = Power & (0b1000000000000000);
214     if( aux == 0x8000){unid = 1.0;}
215     else if(aux == 0){unid = 0.0;}
216
217     frac= Power & (0b0111111111111111);
218     S = (unid + (frac * Vstep))*MxPwr;
219
220
221 }
222
223 void ReactivePower (){
224     /*****
225     /* Transforma uma resposta de 32bits em VoltAmpereReactive
226     /* output variaveis globais Q
227     /*****/
228     uint8 primeiro = 0;
229     uint8 segundo = 0;

```

```

230     uint16 Power = 0;
231     uint16 aux=0;
232     uint16 unid=0;
233     uint16 frac=0;
234     for(uint8 i = 0u; i<5u; i++){
235         //ignorar o primeiro
236         switch(i){
237             case 0: SPIM_1_ReadRxData(); break;    //ingorar o primeiro
238             case 1: primeiro = SPIM_1_ReadRxData(); break;
239             case 2: segundo = SPIM_1_ReadRxData(); break;
240             default:break;
241         }
242
243     }
244     Power = ((segundo<<8) | primeiro);
245
246     aux = Power & (0b1000000000000000);
247     if( aux == 0x8000){unid = 1.0;}
248     else if(aux == 0){unid = 0.0;}
249
250     frac= Power & (0b0111111111111111);
251
252     Q = (unid + (frac * Vstep))*MxPwr;
253
254 }
255
256 void PowerFactor (){
257     /*****
258     /* Transforma uma resposta de 32bits em PowerFactor
259     /* output variaveis globais PF
260     /*****/
261     uint8 primeiro = 0;
262     uint8 segundo = 0;
263     uint16 phi = 0;
264     uint16 aux=0;
265     uint16 unid=0;
266     uint16 frac=0;
267     float neg=0.0;
268     for(uint8 i = 0u; i<5u; i++){
269         //ignorar o primeiro
270         switch(i){
271             case 0: SPIM_1_ReadRxData(); break;    //ingorar o primeiro
272             case 1: primeiro = SPIM_1_ReadRxData(); break;
273             case 2: segundo = SPIM_1_ReadRxData(); break;
274             default:break;
275         }
276
277     }
278     phi = ((segundo<<8) | primeiro);
279

```



```

280     neg = phi & (0b0000010000000000);
281     if( neg == 0x400){neg = -1.0; phi= ~phi;}
282     else if(neg == 0){neg = 1.0;}
283     aux = phi & (0b0000001000000000);
284     if( aux == 0x200){unid = 1;}
285     else if(aux == 0){unid = 0;}
286
287     frac= phi & (0b0000000111111111);
288
289     PF = neg*(unid+(frac*PFstep));
290
291 }
292
293 void customermode(uint8 addr, uint8 p, uint8 s, uint8 t, uint8 q){
294 /*****
295  /*  Envia o codigo de cliente depois de um pequeno delay envia a mensagem
296  /que se pretende escrever na eeprom
297  /*  inputs uint8 addr, uint8 p, uint8 s, uint8 t, uint8 q
298  /(P is least significant byte and Q most significant byte...
299  / out of a 32bit message)
300  /*****/
301     CyDelay(40);
302     SPIrequestData32(0,0x2f,0x6e,0x65,0x70,0x4f);
303     CyDelay(40);
304     SPIrequestData32(0,addr,p,s,t,q);
305 }
306
307 void avg50(){
308 /*****
309  /*  media de 50 leituras
310  /*****/
311
312     float v = 0.0000;    //Tensao RMS
313     float i = 0.0000;    //corrente RMS
314     float p = 0.0000;    //Potência activa
315     float s = 0.0000;    //Potência aparente
316     float q = 0.0000;    //Potência Reactiva
317     float pf = 0.0000;   //Factor de potência
318
319
320     for(uint8 j = 0u; j<50; j++){
321         resetvars();
322         SPIrequestData32(1, Pactive,0,0,0,0);
323         SPIrequestData32(1, Pactive,0,0,0,0);
324         ActivePower();
325         //CyDelay(10);
326
327         SPIrequestData32(1, VrmsIrms,0,0,0,0);
328         SPIrequestData32(1, VrmsIrms,0,0,0,0);
329         V_I_RMS();

```

```

330         //CyDelay(10);
331
332         SPIrequestData32(1, Saparent,0,0,0,0);
333         SPIrequestData32(1, Saparent,0,0,0,0);
334         ApparentPower();
335         //CyDelay(10);
336
337         SPIrequestData32(1, Qreactive,0,0,0,0);
338         SPIrequestData32(1, Qreactive,0,0,0,0);
339         ReactivePower();
340         //CyDelay(10);
341
342         SPIrequestData32(1, PFactor,0,0,0,0);
343         SPIrequestData32(1, PFactor,0,0,0,0);
344         PowerFactor();
345         //CyDelay(10);
346
347         if(VRMs <15 && TRIM == 1){
348             v = v + 0;
349             i = i + 0;
350             p = p + 0;
351             s = s + 0;
352             q = q + 0;
353             pf = pf + 0;
354         }else{
355             v = v + VRMs;
356             i = i + IRMs;
357             p = p + P;
358             s = s + S;
359             q = q + Q;
360             pf = pf + PF;
361         }
362     }
363
364     VRMs = v/50.0;
365     IRMs = i/50.0;
366     P = p/50.0;
367     S = s/50.0;
368     Q = q/50.0;
369     PF = pf/50.0;
370
371 }
372
373
374 int main(void)
375 {
376     CyGlobalIntEnable; /* Enable global interrupts. */
377     //*****I2C*****
378     uint8 userArray[10];
379     uint8 byteCnt=0;

```

```

380     uint8 wrBuf[10] = {0};
381     char rdBuf[10] = {0};
382     char function = '0';
383     I2C_Start(); //inicializacao do bloco i2c
384     //inicializacao do buffer de escrita do I2C Slave<-Master
385     I2C_SlaveInitWriteBuf((uint8 *) wrBuf, 10);
386     //inicializacao do buffer de leitura do I2C Master<-Slave
387     I2C_SlaveInitReadBuf((uint8 *) rdBuf, 10);
388     //*****
389     //**iniciar sensores**
390     //inicializacao do bloco de comunicacao com os sensores de temperatura
391     DS18B20_Start();
392     DS18B20_CheckPresence();
393     //SPI start
394     SPIM_1_Start();
395     VDAC8_1_Start();
396     SPIM_1_EnableRxInt();
397     //*****
398
399     CyDelay(5000);
400     SPIrequestData32(1, VrmsIrms,0,0,0,0);
401     CyDelay(500);
402
403     TRIM=1;
404
405     LED_Write(1);
406
407
408     for(;;)//main loop
409     {
410         //escrita i2c do master
411         //Espera que o master acabe de escrever
412         if(I2C_SlaveStatus() & I2C_SSTAT_WR_CMPLT){
413             //retorna o numero de bytes escritos pelo master
414             byteCnt = I2C_SlaveGetWriteBufSize();
415             //retorna o estado de escrita e dá reset a flag de slave write status
416             I2C_SlaveClearWriteStatus();
417             for(int i=0; i<byteCnt; i++){
418                 //transferir dados do buffer de escrita para um buffer que se possa
419                 //utilizar com segurança
420                 userArray[i]= wrBuf[i];
421             }
422             I2C_SlaveClearWriteBuf();
423             LED_Write( ! LED_Read() );
424             function = userArray[0];
425             // neste caso ao receber qualquer tipo de comunicacao i2c o psoc
426             //irá fazer logo o pedido de temperatura ao sensor
427             if( function == 'x'){DS18B20_SendTemperatureRequest();avg50();}
428             //pedido de temperatura ds18b20 e potencia ACS71020
429         }

```

```
430
431 //leitura i2c do master
432 if(I2C_SlaveStatus() & I2C_SSTAT_RD_CMPLT){
433     //passar a resposta para o buffer de leitura i2c
434     I2C_SlaveClearReadStatus();
435     I2C_SlaveClearReadBuf();
436 }
437 if(DS18B20_DataReady){
438     DS18B20_ReadTemperature();
439     T1 = DS18B20_GetTemperatureAsFloat(0);
440     T2 = DS18B20_GetTemperatureAsFloat(1);
441     T3 = DS18B20_GetTemperatureAsFloat(2);
442     T4 = DS18B20_GetTemperatureAsFloat(3);
443
444 }
445
446 switch(function){
447     case 'l':
448         sprintf(rdBuf, "%.4f", T1 );
449         break;
450         //pedido de leitura ao ds18b20
451     case 'k':
452         sprintf(rdBuf, "%.4f", T2 );
453         break;
454         //pedido de leitura ao ds18b20
455     case 'j':
456         sprintf(rdBuf, "%.4f", T3 );
457         break;
458         //pedido de leitura ao ds18b20
459     case 'h':
460         sprintf(rdBuf, "%.4f", T4 );
461         break;
462         //pedido de leitura ao ds18b20
463     case 'p':
464         sprintf(rdBuf, "%.2f", P);
465         break;
466         //pedido de potencia activa
467     case 'c':
468         sprintf(rdBuf, "%.2f", PF);
469         break;
470         //pedido de factor de potencia
471     case 'y':
472         sprintf(rdBuf, "%.2f", VRMs);
473         break;
474         //pedido de tensao rms
475     case 'i':
476         sprintf(rdBuf, "%.2f", IRMs);
477         break;
478         //pedido de corrente rms
479     case 'x': break;
```

```
480         //comando de pedido de medições sem função nesta parte
481         //Qualquer outro comando que não esteja ainda implementado ou errado
482         default: strcpy(rdBuf, "NEC");
483     }
484
485     }
486 }
487
488 /* [] END OF FILE */
```




CÓDIGO UTILIZADO NO MCU ESP32

Listagem II.1: Código ESP32.

```

1 #include <WiFi.h>
2 #include <Wire.h>
3 #include "AzureIoT.h"
4 #include "Esp32MQTTClient.h"
5
6 #define INTERVAL 1500           //intervalo ente mensagens para o IoT Hub
7 #define DEVICE_ID "esp32"      //ID deste dispositivo na hub
8 #define MESSAGE_MAX_LEN 256    //Tamanho max da mensagem
9
10 #define SDA 21
11 #define SCL 22
12 #define Data_Request_interval 800
13 #define smallWait 50
14 #define swait 5
15 #define SLAVE_ADDR 10
16 #define I2C_msg_size 8
17
18 //Variaveis comunicacao IoT
19 // Please input the SSID and password of WiFi
20 const char* ssid      = "SSID";
21 const char* password = "PASSWORD";
22
23 /*String containing Hostname, Device Id & Device Key in the format: */
24 /*"HostName=<host_name>;DeviceId=<device_id>;SharedAccessKey=<device_key>"*/
25 /*"HostName=<host_name>;DeviceId=<device_id>;
26 SharedAccessSignature=<device_sas_token>"*/
27 static const char* connectionString =
28 "HostName=PcmHubIoT.azure-devices.net;DeviceId=esp32;
29 SharedAccessKey=NBmdusW5qgJETacQgDmdT/J6iigqAq6rjpW5oFo8Lag=";

```

```
30
31 const char *messageData = "{\"deviceId\":\"%s\", \"messageId\":%d,
32 \"Temperature1\":%s, \"Temperature2\":%s, \"Temperature3\":%s,
33 \"Temperature4\":%s, \"ActiveP\":%s, \"PowerFactor\":%s}";
34
35 //const char *messageData="{\"deviceId\":\"%s\", \"messageId\":%d,
36 \"TemperatureS\":%f, \"TemperatureI\":%f}";
37
38 int messageCount = 1;
39 static bool hasWifi = false;
40 static bool messageSending = true;
41 static uint64_t send_interval_ms;
42 //////////////////////////////////////////////////
43 //I2C Communication Variables
44 String responser = "";
45 String responsep = "";
46 String responsec = "";
47 String response1 = "";
48 String response2 = "";
49 String response3 = "";
50 String response4 = "";
51 char x = '0';
52 String responsey = "";
53 String responsei = "";
54 //////////////////////////////////////////////////
55 // Utilities
56 static void InitWifi()
57 {
58     Serial.println("Connecting...");
59     WiFi.begin(ssid, password);
60     while (WiFi.status() != WL_CONNECTED) {
61         delay(500);
62         Serial.print(".");
63     }
64     hasWifi = true;
65     Serial.println("WiFi_connected");
66     Serial.println("IP_address:");
67     Serial.println(WiFi.localIP());
68 }
69
70 static void SendConfirmationCallback(IOTHUB_CLIENT_CONFIRMATION_RESULT result)
71 {
72     if (result == IOTHUB_CLIENT_CONFIRMATION_OK)
73     {
74         Serial.println("Send_Confirmation_Callback_finished.");
75     }
76 }
77
78 static void MessageCallback(const char* payLoad, int size)
79 {
```

```

80     Serial.println("Message_callback:");
81     Serial.println(payload);
82 }
83
84 static void DeviceTwinCallback(DEVICE_TWIN_UPDATE_STATE updateState,
85 const unsigned char *payload, int size)
86 {
87     char *temp = (char *)malloc(size + 1);
88     if (temp == NULL)
89     {
90         return;
91     }
92     memcpy(temp, payload, size);
93     temp[size] = '\0';
94     // Display Twin message.
95     Serial.println(temp);
96     free(temp);
97 }
98
99 static int DeviceMethodCallback(const char *methodName,
100 const unsigned char *payload, int size,
101 unsigned char **response, int *response_size)
102 {
103     LogInfo("Try_to_invoke_method_%s", methodName);
104     const char *responseMessage = "\"Successfully_invoke_device_method\"";
105     int result = 200;
106
107     if (strcmp(methodName, "start") == 0)
108     {
109         LogInfo("Start_sending_temperature_and_humidity_data");
110         messageSending = true;
111     }
112     else if (strcmp(methodName, "stop") == 0)
113     {
114         LogInfo("Stop_sending_temperature_and_humidity_data");
115         messageSending = false;
116     }
117     else
118     {
119         LogInfo("No_method_%s_found", methodName);
120         responseMessage = "\"No_method_found\"";
121         result = 404;
122     }
123
124     *response_size = strlen(responseMessage) + 1;
125     *response = (unsigned char *)strdup(responseMessage);
126
127     return result;
128 }
129 //funcoes criadas individualmente para evitar

```

```
130 //perda de tempo
131 static void Data_Request(char f, int addr){
132     Wire.beginTransmission(addr);
133     Wire.write(f);
134     Wire.endTransmission();
135 }
136 static void Data_Readl(int addr){
137     Wire.requestFrom(addr,I2C_msg_size);
138     while( Wire.available() ){
139         x = Wire.read();
140         response1 += x;
141     }
142 }
143 static void Data_Readk(int addr){
144     Wire.requestFrom(addr,I2C_msg_size);
145     while( Wire.available() ){
146         x = Wire.read();
147         response2 += x;
148     }
149 }
150
151 static void Data_Readj(int addr){
152     Wire.requestFrom(addr,I2C_msg_size);
153     while( Wire.available() ){
154         x = Wire.read();
155         response3 += x;
156     }
157 }
158 static void Data_Readh(int addr){
159     Wire.requestFrom(addr,I2C_msg_size);
160     while( Wire.available() ){
161         x = Wire.read();
162         response4 += x;
163     }
164 }
165 static void Data_Readp(int addr){
166     Wire.requestFrom(addr,I2C_msg_size);
167     while( Wire.available() ){
168         x = Wire.read();
169         responsep += x;
170     }
171 }
172 static void Data_Readc(int addr){
173     Wire.requestFrom(addr,I2C_msg_size);
174     while( Wire.available() ){
175         x = Wire.read();
176         responsec += x;
177     }
178 }
179 static void Data_Readv(int addr){
```

```

180     Wire.requestFrom(addr,I2C_msg_size);
181     while( Wire.available() ){
182         x = Wire.read();
183         responsey += x;
184     }
185 }
186 static void Data_Read1(int addr){
187     Wire.requestFrom(addr,I2C_msg_size);
188     while( Wire.available() ){
189         x = Wire.read();
190         responsei += x;
191     }
192 }
193 ////////////////////////////////////////////////////
194 // Arduino sketch
195 void setup()
196 {
197     Serial.begin(115200);
198     Serial.println("ESP32_Device");
199     Serial.println("Initializing...");
200
201     // Initialize I2C
202     Wire.begin(SDA,SCL);
203
204     // Initialize the WiFi module
205     Serial.println("<>WiFi");
206     hasWifi = false;
207     InitWifi();
208     if (!hasWifi)
209     {
210         return;
211     }
212     randomSeed(analogRead(0));
213
214     Serial.println("<>IoT_Hub");
215     Esp32MQTTClient_SetOption(OPTION_MINI_SOLUTION_NAME , "GetStarted");
216     Esp32MQTTClient_Init((const uint8_t*)connectionString, true);
217
218     Esp32MQTTClient_SetSendConfirmationCallback(SendConfirmationCallback);
219     Esp32MQTTClient_SetMessageCallback(MessageCallback);
220     Esp32MQTTClient_SetDeviceTwinCallback(DeviceTwinCallback);
221     Esp32MQTTClient_SetDeviceMethodCallback(DeviceMethodCallback);
222
223     send_interval_ms = millis();
224 }
225
226 void loop()
227 {
228     if (hasWifi)
229     {

```

```
230     if (messageSending && (int)(millis() - send_interval_ms) >= INTERVAL)
231     {
232         //Data Request from slave
233         Data_Request('x', SLAVE_ADDR);
234         delay(Data_Request_interval);
235
236         Data_Request('l', SLAVE_ADDR);
237         delay(smallWait);
238         Data_Readl(SLAVE_ADDR);
239         delay(swait);
240         Data_Request('k', SLAVE_ADDR);
241         delay(smallWait);
242         Data_Readk(SLAVE_ADDR);
243         delay(swait);
244         Data_Request('j', SLAVE_ADDR);
245         delay(smallWait);
246         Data_Readj(SLAVE_ADDR);
247         delay(swait);
248         Data_Request('h', SLAVE_ADDR);
249         delay(smallWait);
250         Data_Readh(SLAVE_ADDR);
251         delay(swait);
252         Data_Request('p', SLAVE_ADDR);
253         delay(smallWait);
254         Data_Readp(SLAVE_ADDR);
255         delay(swait);
256         Data_Request('c', SLAVE_ADDR);
257         delay(smallWait);
258         Data_Readc(SLAVE_ADDR);
259         delay(swait);
260         Data_Request('y', SLAVE_ADDR);
261         delay(smallWait);
262         Data_Readv(SLAVE_ADDR);
263         Data_Request('i', SLAVE_ADDR);
264         delay(smallWait);
265         Data_Readi(SLAVE_ADDR);
266
267         //opcional serve apenas para testes e debug
268         Serial.print("ReceivedT1:");
269         Serial.println(response1);
270         Serial.print("ReceivedT2:");
271         Serial.println(response2);
272         Serial.print("ReceivedT3:");
273         Serial.println(response3);
274         Serial.print("ReceivedT4:");
275         Serial.println(response4);
276         Serial.print("ReceivedP:");
277         Serial.println(responsep);
278         Serial.print("ReceivedC:");
279         Serial.println(responsec);
```

```

280     Serial.print("ReceivedVrms:");
281     Serial.println(responsey);
282     Serial.print("ReceivedIrms:");
283     Serial.println(responsei);
284     //*****
285     // Envio dos dados recebidos por i2c
286     char messagePayload[MESSAGE_MAX_LEN];
287     snprintf(messagePayload,MESSAGE_MAX_LEN, messageData, DEVICE_ID,...
288     messageCount++, response1, response2, response3, response4,...
289     responsey, responsei, responsep, responsec);
290     Serial.println(messagePayload);
291     EVENT_INSTANCE* message = ...
292     Esp32MQTTClient_Event_Generate(messagePayload,MESSAGE);
293     Esp32MQTTClient_Event_AddProp(message, "temperatureAlert", "true");
294     Esp32MQTTClient_SendEventInstance(message);
295     response1 = "";
296     response2 = "";
297     response3 = "";
298     response4 = "";
299     responsey = "";
300     responsep = "";
301     responsec = "";
302     responsei = "";
303     x='0';
304     send_interval_ms = millis();}
305     else{    Esp32MQTTClient_Check(); } }
306     delay(10);}

```




Query UTILIZADA NO SERVIÇO *Stream Analytics*

Listagem III.1: Query utilizada no serviço *Stream Analytics*.

```
1 SELECT
2     CAST(iothub.EnqueueedTime AS datetime) AS TimeStamp,
3     CAST(messageId AS bigint ) AS MessageID,
4     CAST(Temperature1 AS float) AS Temperature1_DS18B20,
5     CAST(Temperature2 AS float) AS Temperature2_DS18B20,
6     CAST(Temperature3 AS float) AS Temperature3_DS18B20,
7     CAST(Temperature4 AS float) AS Temperature4_DS18B20,
8     CAST(Vrms as float) AS VoltageRMS,
9     CAST(Irms as float) AS CurrentRMS,
10    CAST(ActiveP AS float) AS ActivePower,
11    CAST(PowerFactor AS float) AS PowerFactor
12 INTO IotHubOutput
13 FROM IoTHubInput
14
15 SELECT
16     CAST(iothub.EnqueueedTime AS datetime) AS TimeStamp,
17     CAST(messageId AS bigint ) AS MessageID,
18     CAST(Temperature1 AS float) AS Temperature1_DS18B20,
19     CAST(Temperature2 AS float) AS Temperature2_DS18B20,
20     CAST(Temperature3 AS float) AS Temperature3_DS18B20,
21     CAST(Temperature4 AS float) AS Temperature4_DS18B20,
22     CAST(Vrms as float) AS VoltageRMS,
23     CAST(Irms as float) AS CurrentRMS,
24     CAST(ActiveP AS float) AS ActivePower,
25     CAST(PowerFactor AS float) AS PowerFactor
26 INTO blobstorageoutput
27 FROM IoTHubInput
```